





UNIVERSITÀ  
POLITECNICA  
DELLE MARCHE

# Master Thesis

Master Program in Biomedical Engineering

Università Politecnica delle Marche

Title: NLP based chatbot for The IA Institute

From April 2019 to September 2019

KAGMENI FOSSI Alex Frank

Academic supervisor: Lorenzo Scalise & Saïd Moussaoui

## ABSTRACT

---

Artificial intelligence and conversational agents have become increasingly important in recent years due to technological progress. Chatbots are attracting growing interest, especially in companies. The IA Institute is also dealing with the subject of chatbots and examines products of various customers, in order to improve the student services by introducing them. Therefore, this study investigates the suitability of an open source chatbots solution of the manufacturer Rasa Technologies GmbH for the service provision of the company in the areas of Student Services and Admissions Services.

By reading scientific reports, reference books, and the manufacturer's technical documentation, knowledge was acquired which served to understand methodical procedures and to apply them correctly. This included collecting and recording training data for the models, developing and training a chatbot prototype and implementing it into an existing test environment.

The most important finding was that a chatbot's responses reflected the training it had received. The quality of the collected data, as well as its amount, are again the prerequisite for good training.

The investigations showed that Rasa's open source product can be considered as a chatbot solution for frequently asked questions. The installation, training, and implementation in an existing environment was feasible and the answers to the questions worked quite well. If the chatbot should be able to manage longer conversational flows, it needs a correspondingly higher training effort, which grows with the increasing amount of training data.

**Keywords:** Artificial Intelligence, Chatbot, Machine Learning, NLP, Rasa

# CONTENTS

INTRODUCTION .....	1
ARTIFICIAL INTELLIGENCE AND CHATBOTS .....	2
Artificial Intelligence.....	2
Types of Artificial Intelligence .....	5
History of Artificial Intelligence .....	6
Machine learning.....	8
Artificial neural network .....	10
Natural Language Processing .....	13
Natural Language Understanding .....	16
Chatbots .....	17
History of chatbots.....	18
Current state of chatbots .....	19
Future of chatbots.....	20
Need for chatbots and their benefits for companies and users .....	22
Terminologies used .....	23
Types of chatbots .....	25
Processing of a chatbot .....	27
Training and testing.....	28
METHODS .....	32
Project management.....	32
Process model .....	33
Meetings.....	33
Project boundaries and system delimitation .....	33
Prototype.....	33
Target group.....	34
Data collection.....	34
Scope of data.....	34
Data collection .....	34
Survey.....	35
Brainstorming and brainwriting .....	35
Interview .....	36
Real conversation .....	36
Consultation of experts.....	37
DATA COLLECTION AND TRAINING DATA .....	38

Results .....	38
Survey .....	38
Brainstorming and brainwriting .....	39
Interview .....	40
Evaluation of training data .....	41
Manners .....	41
Accommodation .....	42
University .....	42
Leisure time .....	42
Administrative tasks .....	43
TECHNOLOGIES AND TOOLS.....	44
Linux .....	44
Apache Hypertext Transfer Protocol Server .....	44
Python .....	45
Anaconda Distribution .....	45
Rasa Stack.....	46
Rasa NLU .....	46
Rasa Core.....	46
Rasa Platform .....	47
CHATBOT DEVELOPMENT AND IMPLEMENTATION.....	48
Software installation .....	49
NLU model.....	50
Data format .....	51
Data entering.....	52
Synonyms .....	53
NLU model configuration .....	54
Pipeline.....	55
NLU model training .....	55
NLU model testing.....	56
Dialogue management model.....	57
Domain format .....	58
Tone of voice .....	59
Story data format .....	61
Online training.....	62
Creating chatbot.....	66
Chatbot on the Linux command line .....	66
Chatbot on the own website.....	67

Chatbot operation.....	69
Logging .....	69
Negation and misspelling.....	70
Fallback action.....	71
Training process .....	71
Migration guide.....	72
DEVELOPMENT FINDINGS AND RESULTS .....	73
Data .....	73
Installation and development .....	73
Operation and periodic training.....	75
Testing.....	76
CONCLUSION AND RECOMMENDATION.....	77
AUTHOR’S REFLECTION .....	79
REFERENCES .....	80

## APPENDICES

Appendix 1	SURVEY ON INFORMATION RETRIEVAL AT The AI Institute
Appendix 2	SYSTEM AND SOFTWARE INFORMATION
Appendix 3	INSTALLED PYTHON PACKAGES
Appendix 4	PYTHON SCRIPT – NLU_MODEL.PY
Appendix 5	PYTHON SCRIPT – TRAIN_INIT.PY
Appendix 6	PYTHON SCRIPT – TRAIN_ONLINE.PY
Appendix 7	PYTHON SCRIPT – DIALOGUE_MGMT_MODEL.PY

## FIGURE CONTENT

Figura 1 Definitions of Artificial Intelligence, organized in four categories (Russell & Norvig, 2010, p. 1) .....	4
Figura 2 Schematic view of an artificial neuron within an artificial neural network (Castrounis Alex, 2016; Traeger et al., 2003, p. 1056) .....	11
Figura 3 Schematic view of an artificial neural network (Khademi & Jamal, 2016, p. 3; Sayadi, Monjezi, Talebi, & Khandelwal, 2013, p. 320).....	12
Figura 4 Constituency structure after parsing a sentence (van Duin & Bakhshi, 2018, p. 15)....	16
Figura 5 Demarcation between NLP and NLU (Olivier, 2017) .....	17
Figura 6 Chatbot trends of the last five years based on web search term (Raj, 2019, p. 2) .....	19
Figura 7 Size of the chatbot market worldwide, in 2016 and 2025 (Statista GmbH, 2017b) .....	21
Figura 8 Size of the chatbot market worldwide, in 2017 and 2024 (Statista GmbH, 2018) .....	22
Figura 9 Illustration of the three most important terms using a sentence (McGrath, 2017) .....	24
Figura 10 Classification of chatbots (Nimavat & Champaneria, 2017, p. 1020) .....	25
Figura 11 Survey results: The use of a chatbot .....	39
Figura 12 . Dialogue handling using Rasa Core (Petraityte, 2018b, p. 20) .....	47
Figura 13 Rasa Platform diagram (Rasa Technologies GmbH, 2019b).....	48
Figura 14 Intent classification and entity extraction (Petraityte, 2018a) .....	50
Figura 15 Entering training data using the Rasa NLU Trainer .....	52
Figura 16 . Entering training data in Linux terminal.....	53
Figura 17 Using synonyms for training data.....	54
Figura 18 Defining the NLU model configuration .....	55
Figura 19 Successful training of the NLU model .....	56
Figura 20 List of needed Rasa NLU model files .....	56
Figura 21 Successful testing of the NLU model.....	57
Figura 22 Rasa Core predicts next actions and steps (Petraityte, 2018a).....	57
Figura 23 Examples of definitions within the domain.....	59
Figura 24 Most important characteristic of a chatbot .....	59
Figura 25 Preferred personality of a chatbot by country (Statista GmbH, 2017a) .....	60
Figura 26 Examples of stories within the story file .....	61
Figura 27 Successful training of the dialogue management model.....	62
Figura 28 Successful running of interactive learning .....	63
Figura 29 Training of the dialogue management model by interactive learning.....	63
Figura 30 Selecting action after training .....	64
Figura 31 Provide the files in which the training should be saved.....	64
Figura 32 Visualization of the conversation during interactive learning .....	65
Figura 33 Visualization of the conversational paths of a story file .....	66
Figura 34 Chat with a chatbot inside the Linux terminal .....	67
Figura 35 Embedding the chatbot widget on the website.....	68
Figura 36 Chat with a chatbot implemented on a website.....	69
Figura 37 Make log data readable using a JSON formatter .....	70
Figura 38 Fallback action of the chatbot.....	71



## TABLE CONTENT

Table 1 Microsoft's confidence level examples for the cloud-based solution LUIS (Microsoft Corporation, 2018) .....	25
Table 2 The seven categories of chatbot's design (Muñoz-Romero et al., n.d., 2017).....	31

## LIST OF ABBREVIATIONS

AI	Artificial Intelligence
AIML	Artificial Intelligence Markup Language
AGI	Artificial General Intelligence
ANI	Artificial Narrow Intelligence
API	Application Programming Interface
ASI	Artificial Super Intelligence
CMS	Content Management System
CSS	Cascading Style Sheets
FAQ	Frequently Asked Questions
HMM	Hidden Markov Model
HTTP	Hypertext Transfer Protocol
IoT	Internet of Things
IT	Information Technology
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
LUIS	Language Understanding Intelligent Service
NER	Named Entity Recognition
NLP	Natural Language Processing
NLU	Natural Language Understanding
POS	Part-of-Speech
REST	Representational State Transfer
TCP	Transmission Control Protocol
XML	Extensible Markup Language

## INTRODUCTION

Along with Blockchain and Internet of Things (IoT), Artificial Intelligence (AI) has been one of the most interesting trends in the world of Information Technology (IT) for the past two years. Especially, AI and machine learning have reached a critical point and, according to Gartner Inc. (2016), they will increasingly complement and extend virtually any technology-based service, application or thing. They are already being used in various areas today and there are many indications that their field of application will expand in the future. (Panetta, 2016, 2017, 2018)

Gartner's trends also include AI-based conversation systems and platforms that range from simple informal, bidirectional text and voice conversations to more complex interactions. An example of such conversation systems are chatbots. Since the initial attempts of ELIZA and ALICE, which are considered to be the ancestors of the chatbots known today, many intelligent bots have been developed and used for a wide variety of applications. In addition to the various ways in which chatbots can be used, companies have discovered the benefits of using them, especially in the area of process optimization. The AI Institute should also benefit from some of these advantages, which is why this thesis is tackled. (Klopfenstein, Malatini, & Bogliolo, 2017, pp. 556–557; Mornhinweg, 2018; Panetta, 2016, 2017, 2018; Sahin, Relieu, & Francillon, 2017)

The aim of the thesis is to investigate the suitability of an open source chatbot solution from the manufacturer Rasa Technologies GmbH for the service provision of The AI Institute in the field of Student Services and Admissions Services. The target group of the product to be investigated consists initially of international students, and locals who want to do their studies in a short period of times. In order to ensure a qualitative implementation of this research, answers to five research questions are determined with the help of external theory and self-applied practice.

This thesis deals with the following research questions:

- How can the chatbot be taught for the use in The AI Institute?
- How good are the answers of the chatbot to the questions asked by the user?
- Under what circumstances can the chatbot be set up in an existing environment?
- How are the chatbot answers related to the information provided by The AI Institute on their website?
- How can the administrative effort be reduced by the chatbot operation?

## ARTIFICIAL INTELLIGENCE AND CHATBOTS

In this chapter, the theoretical basic knowledge relevant for the present thesis is explained by research of literature. In the beginning, an overview of the topic AI, its history and its subareas is given. This is followed by a theoretical introduction to the topic of chatbots.

### Artificial Intelligence

**Artificial intelligence (AI)**, is the ability of a digital computer or computer-controlled robot to perform tasks commonly associated with intelligent beings. A milestone for the definition of the intelligence of machines, however, has been the Turing Test since 1950, also known as imitation game. Turing considered that a machine could be described as intelligent if an average interviewer when asking questions to a machine that is not visible to him and a person that is not visible to him, could not tell from the answers received which of the two interviewees is the machine and which is the person. After a few adjustments to the imitation game and the detections of another variant of the game, Turing was convinced that computers could be programmed to play the imitation game successfully, provided that they had sufficient computing speed and memory capacity. Turing explained that a machine or a computer would be considered intelligent if it were able to imitate a person's ability to communicate in natural language. The prerequisite for this would be to ensure permanent and trial-independent deception. Since the ability to deal with natural language is closely linked to thinking and intelligence, the imitation game

finally became the intelligence test, the so-called Turing Test. Turing's assessments led to contradictions from many different directions. However, the discussion about the Turing Test has not been torn down since its publication in 1950. While some see the starting point of the AI in the Turing Test, others regard it as misleading and completely inadequate as an intelligence test. 50 years after Turing, there is still no agreement in AI research on the validity of the Turing Test, but it seems to serve the developers of chatbots as a significant orientation. (A. Braun, 2003, p. 22; Döbel et al., 2018, p. 44; Storp, 2002, pp. 6–7; Turing, 1950; Turkle, 1995, pp. 85–86)

The Turing Test was never realized in its original form. Instead, the Loebner Contest was founded in 1990 by Hugh Loebner due to the emergence of more powerful computers, which made practical testing possible. The aim of the annual contest is to pass a series of tasks of a modified Turing Tests. Chatbot solutions that pass this test receive a gold medal and prize money of 100,000 US dollars. Bronze medals and 2,000 US dollars are also awarded for the most human-like program. So far only bronze medals have been awarded. (Storp, 2002, pp. 7–9)

Figure 1 shows eight definitions of AI, which are laid out along two dimensions. The definitions on the bottom are concerned with behavior, whereas the ones on the top address thought processes and reasoning. The definitions on the right measure against an ideal measure of performance, called rationality. A system is called rational if it does the right thing based on what it knows. (Russell & Norvig, 2010, p. 1) "The definitions on the left measure success in terms of fidelity to human performance" (Russell & Norvig, 2010, p. 1).

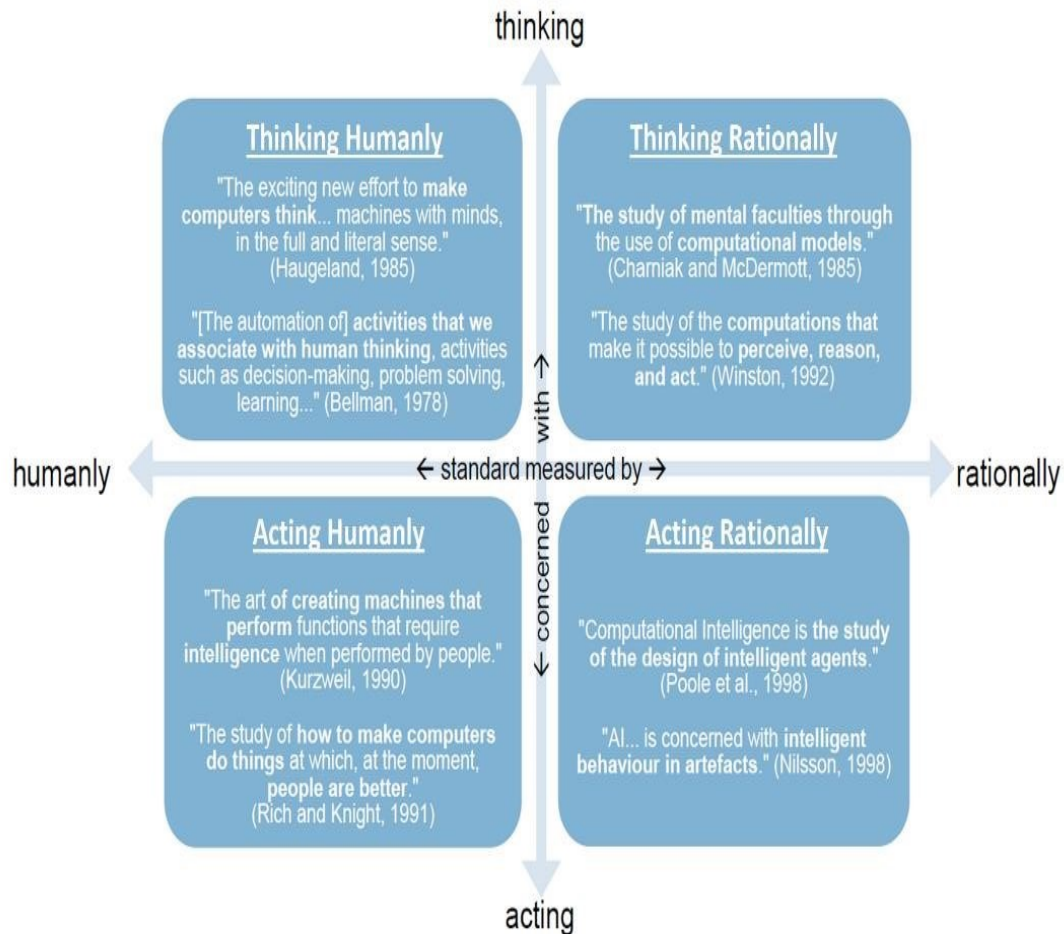


Figura 1 Definitions of Artificial Intelligence, organized in four categories (Russell & Norvig, 2010, p. 1)

Historically, each of the four AI approaches shown in the figure has been followed by different people using different methods. A rationalistic approach consists of a combination of mathematics and engineering. A human-centered approach has to consist in part of empirical science that includes hypotheses and observations about human behavior. The groups, with their different approaches, have helped and denigrated each other. (Russell & Norvig, 2010, pp. 1–2) So when one dives into them, the techniques used in AI are just advanced forms of mathematical and statistical models. All these cleverly designed models provide tools for computing tasks that were previously reserved for humans. (van Duin & Bakhshi, 2018, p. 15)

## Types of Artificial Intelligence

AI can be classified into three types according to its level of autonomy. (Scherk, Pöchhacker-Tröscher, & Wagner, 2017, p. 20)

The term Artificial Narrow Intelligence (ANI), also known as weak AI, refers to the most limited form of AI and the current state of the art. ANI can perform specific tasks with a combination of machine learning, complex algorithms, and numerous other techniques depending on the application area. This AI is called narrow because it is only able to perform a task within its specific purpose. When interacting with people, ANI often uses Natural Language Processing (NLP) techniques. ANI relies on machine learning algorithms that require huge data sets and adaptation of our actual world to understand them. Current ANI systems can only perform their tasks because the human's natural environment was already adapted for them by humans. The innovations around ANI for the next 25 years are limited to the execution of certain tasks or a defined group of specific tasks. (Deloitte Touche Tohmatsu Limited, 2016, pp. 2, 4)

The term Artificial General Intelligence (AGI), also known as human-level AI or strong AI, describes a level of AI that would be able to perform any task according to a human standard. When the time of the AGI will be reached is still unclear and is debated. However, most specialists in the field of AI are of the opinion that this will be around the year 2040. To reach the AI level AGI, hardware comparable to the human brain is needed that can perform about ten billion calculations per second with only 20 watts of power. Such a computer already exists in Chinese Tianhe-2, which can perform 34 quadrillion calculations per second. But Tianhe-2 is clearly not suitable for the consumer market due to its size of 720 square meters, its cost of 390 million US dollars and its total power of 24 megawatts. This shows, however, that the hardware is not the biggest problem in the development of an AGI, but the difficulties lie much more in the area of software. The challenge is to develop a software that is as advanced as a human's mind. At present, such software does not exist, but the majority of computer scientists assume that the most promising method for programming an AI is to program an AI that can change itself by writing its own codes. This would allow it to continually improve, become more intelligent and build on its own intelligence. (Deloitte Touche Tohmatsu Limited, 2016, pp. 2–4) .

When talking about Artificial Super Intelligence (ASI), it means an AI that is a level higher than AGI and much more advanced than human intelligence. Because of the law of accelerating returns, ASI could potentially be realized in few days after AGI is achieved. Consequences of the development of ASI are so far unknown. The level of intelligence that ASI would have would be 4 billion times more intelligent than the most intelligent human being. It is therefore not possible to predict what the effects of such a technological lead will be. A cure for cancer, human mortality, eternal timelessness or the extinction of mankind are possible events that could happen. While Stephen Hawking associates such a development with the end of mankind, Ray Kurzweil thinks that it would be possible to introduce a human-AI synthesis with it. Although this may sound a little absurd, the impact of the ANI in 2016 on the labor force can be seen. It is even estimated that half of the jobs in 2050 will be managed by AI. (Deloitte Touche Tohmatsu Limited, 2016, p. 3) .

### History of Artificial Intelligence

What is known today as AI has its origin in the 1940s through Warren McCulloch and Walter Pitt, who investigated the theory of artificial neural networks. They showed that any computable function can be calculated using a network consisting of connected neurons (Döbel et al., 2018, p. 9; Russell & Norvig, 2010, p. 16). In 1956, ten US researchers, employed by IBM, the Massachusetts Institute of Technology, Princeton University, and other institutions sat together to conduct a planned 2-month workshop on automata theory, artificial neural networks, and intelligence for a study on AI. The workshop which took place at Dartmouth College did not bring any new breakthroughs in AI, but it did ensure that the most important figures now know each other. (Russell & Norvig, 2010, pp. 17–18). Between 1952 and 1969, the work of various AI researchers flourished, but only to a limited extent. Although it was initially thought that computers and their programs were capable of many things, it soon became clear that they were only arithmetical in nature. Although many different approaches and AI programs were developed, such as the various programs by Arthur Samuel, which dealt with playing checkers, the intellectual establishment gradually realized that a machine could not do many things. Therefore, the statement “a machine can never do X” became well known and regularly supplemented with new terms for X. (Russell & Norvig, 2010, pp. 18–20)

The examples listed for X by Turing are “Be kind, resourceful, beautiful, friendly, have initiative, have a sense of humor, tell right from wrong, make mistakes, fall in love, enjoy strawberries and cream ...” (Russell & Norvig, 2010, p. 1021).

Between 1966 and 1973, the AI world and the AI systems struggled with three difficulties due to exaggerated predictions of their progress. The first type of difficulty was that most early programs knew nothing of the subject they were working on. For example, translations that can only be done by syntactic manipulations. The fact is, however, that accurate translation needs background knowledge to solve ambiguities and determine the sentence’s content. The second type of difficulty was the persistence of many problems that the AI tried to solve. Problems were solved by most early AI programs by trying different step combinations until the solution was found. This strategy initially only worked because microworlds had very few objects. So there were very short solution sequences and very few possible actions. The third difficulty resulted from some basic limitations of the fundamental structures used to create intelligent behavior. (Russell & Norvig, 2010, pp. 20–22)

After no real success for a long time, it was only in the 1980s that AI was given a place in the industry due to the introduction of expert systems and the humanoid robotics. The AI industry was booming. From several millions of dollars in 1980 to billions of dollars in 1988. There were hundreds of companies that developed robots, expert systems, vision systems, and hardware and software at that time. Shortly thereafter the second so-called AI winter occurred, whereupon it became calmer again. (Döbel et al., 2018, pp. 9–10; Russell & Norvig, 2010, p. 24)

After the research on artificial neural networks stagnated in the period from 1974 to 1980 due to too slow computers, it gained a foothold in science again in 1986. The reason for the return of artificial neural networks was the reinvention of the back-propagation learning algorithm found by Bryson and Ho in 1969. This remained so until 1995, but from then on learning methods such as support vector machines turned out to be manageable for practical applications. (Döbel et al., 2018, pp. 9–10; Russell & Norvig, 2010, pp. 24–25)



In 1996 IBM developed the Deep Blue, which won against the chess world champion Kasparov with the help of AI. Around the turn of the millennium, advances in computer technologies and the emergence of Big Data made it possible to learn very complex, so-called deep artificial neural networks. This was the beginning of the success of today's AI. This was followed by further developments such as the IBM Watson or the Alpha Go, which won against humans in complex games. (Döbel et al., 2018, pp. 9–10)

### Machine learning

Machine learning is in computer science one of the fastest growing areas with applications that are far-reaching. As an area of AI, machine learning aims to create “knowledge” from “experience” by using learning algorithms that develop a complex model using examples. This generated model and the automatically acquired knowledge representation can then be applied to new and potentially unknown data of the same type. Machine learning can be used in areas where processes are too complicated for an analytical description, but there is a sufficient amount of example data available. With the models learned, decisions and recommendations can be generated or predictions can be made. The powerful performance of machine learning-based systems is no indication that the machine being executed has any awareness or understanding of what data it is processing, for what reason it is doing this, in what context, or what meaning the data has. (Döbel et al., 2018, p. 8; Shalev-Shwartz & Ben-David, 2014, p. 3; Wrobel, Joachims, & Morik, 2013, p. 405) Compared to classical programming, machine learning poses different challenges. In principle, the rule is that with a larger amount of training data the learning algorithm receives, much better models can be created and its error rate can be reduced. The challenge now is to keep the model generic enough to work equally well with new data that differs from the training data and did not occur during the training phase. In addition, the models should show a certain robustness, so that similar inputs also deliver similar results. The quality of the generated model depends on the quality of the training data. If the algorithm is trained with too many wrong examples, it cannot learn the correct answer. It can also be possible that when using unrepresentative training data, the outputs of the learning algorithm for new types of inputs are subject to greater uncertainty. There are also models that, in addition to their output, also provide estimates of how reasoned each output is. (Döbel et al., 2018, pp. 11–12)

Depending on the intended use, there are less or more suitable learning styles for machine learning methods. A machine learning method is a procedure that can be realized by several concrete learning algorithms. The learning styles are distinguished from each other by additional information and the learning tasks that are possible as a result. The most important learning styles include supervised learning, unsupervised learning and reinforcement learning.

Supervised learning works with so called labels. These are the correct answers to the example data, which must be provided. This usually results in more work for data preparation. But it is necessary for the classification of objects and the prediction or estimation of values.

Unsupervised learning has no labels and no feedback, only raw example data to discover basic patterns. For example, by finding simplifying descriptions of the entire set of examples. In this way, data can be divided into clusters or different groups, or the number of characteristics can be reduced.

In reinforcement learning, the learning algorithms and machines receive feedback from their interactions with the environment. This allows future actions to be improved and errors to be reduced. Such a kind of learning is often found in robotics. An example of this would be learning the best gripping movement for objects. (Döbel et al., 2018, pp. 10, 43, 45, 46)

Areas of application of machine learning-based scientific publications, patent applications or public research projects are image and video analysis, audio processing, text and speech processing, signal processing and heterogeneous data. Image and video analysis is the processing of visual data that enables a machine to perceive and identify scenes, activities, and objects in the environment. Text and language processing includes methods and techniques that enable computer systems to interpret and produce natural language in word and writing, also known as NLP. (Döbel et al., 2018, pp. 13, 22–23)

Due to the development of a machine learning based chatbot, the field of application of this work is word processing and the learning style is supervised learning.

## Artificial neural network

Artificial neural networks are models of machine learning. These networks, which are inspired by aspects of the human brain, consist of many layers of nodes realized in software, which are called artificial neurons. These artificial neurons are interconnected and have numerical values that are regarded as weights on these connections. To train such an artificial neural network, the weights of the neuron connections are changed by a learning algorithm until the results are good enough for the task. (Döbel et al., 2018, p. 44)

Artificial neural networks attempt to simulate the central nervous system processes of higher organisms by simulating the behavior of individual neurons and their interaction with each other using special software. In contrast to the classical static calculation methods, artificial neural networks have a learning ability. More precisely, this means that an artificial neural network has no information at the beginning. This information is only extracted through the application of a certain number of known examples. Ideally, an artificial neural network practiced through repeated training can generalize. This means changing the connections within the neural network in such a way as to ensure that the artificial neural network correctly classifies unknown data according to the learned rules. When data is incomplete or disturbed, artificial neural networks are not very sensitive. They have also shown in various areas that they are suitable for modeling complex time-dependent systems and predicting events. (Traeger et al., 2003, p. 1057)

There are many types of artificial neural networks. However, they can all be defined by three peculiarities. According to Traeger et al. (2003, p. 1055), Lawrence (1992), Zell (1996), these are the individual neurons, their connections, which together with the neurons form the architecture, and the learning rule within the network. (Traeger et al., 2003, p. 1055)

The neurons, which can be described at a certain point in time by their activation state, form the basic building blocks of an artificial neural network. The activation function in the neuron determines what the new state will be like. This, in turn, results from the currently incoming new information and the already existing activation. The output function determines whether an activation of the neuron takes place based on the summed stimuli. The connections between the neurons vary in strength and are called weights. The sum of the inputs to a neuron is formed from the

outputs of the preceding neurons and the respective connection weights and results in the activation function. If a certain threshold value is exceeded, the neuron, in turn, transmits a signal. Activation functions manage the behavior of the neuron and ensure that if the excitation of the neuron is low, the neuron will not discharge. In the same way, it also triggers the activity of a neuron as soon as a critical limit is exceeded by an increase in input. Figure 2 shows the schematic view of an artificial neuron within an artificial neural network. (Traeger et al., 2003, pp. 1055–1056)

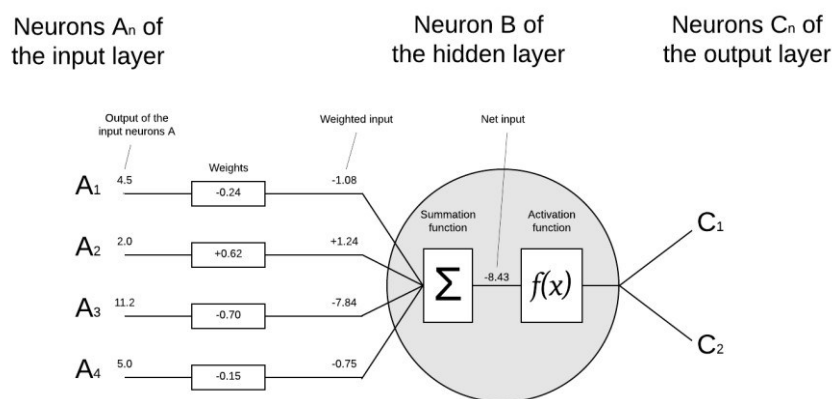


Figura 2 Schematic view of an artificial neuron within an artificial neural network (Castrounis Alex, 2016; Traeger et al., 2003, p. 1056)

The input of the neuron corresponds to the information flow and the output of the signal forwarding. The incoming pulses are summed up inside the neuron. The activation function determines from which threshold value an output is to take place. The neuron B shown in figure 2 receives the value 2.0 as input, from the output of the neuron A<sub>2</sub>, which is then multiplied by the weight 0.62. The discharge of neuron B to neurons C<sub>1</sub> and C<sub>2</sub> only takes place when the sum of the inputs in neuron B reaches the threshold value of the activity function. The values and weights within the figure are only examples. (Traeger et al., 2003, p. 1056)

Artificial neural networks consist of numerous artificial neurons and their connections. There are three different types of neurons. First, there are the input neurons, which receive external stimuli or variables that are to be processed. The second type are the output neurons, which represent the results. The third type of neurons are the hidden neurons. These are hidden between the input and output neurons and receive impulses, process them

and finally pass them on. In most cases, the number of input neurons corresponds to the number of input variables. The information of the neurons from the input layer is forwarded at least to one hidden layer. Neurons of one layer are structurally linked to each neuron of the next layer. A connection between the neurons can be functionally interrupted. This is the case, for example, if the weighting of a connection is exactly zero. Depending on the problem to be solved, the number of layers and neurons in the artificial neural networks varies. The challenge is to define the number large enough so that tolerable results can be achieved but not too large so that the ability to generalize is not impaired. So far, there is no deterministic approach to define an optimal network structure for neural networks. Thus, in practice, several network topologies must always be tested for their functionality. Traeger et al. (2003, p. 1057), Lawrence (1992) mentioned a rule of thumb which says that the number of hidden neurons should be half the sum of output and input neurons to achieve good performance. Figure 3 shows the schematic view of an artificial neural network. (Traeger et al., 2003, pp. 1056–1057)

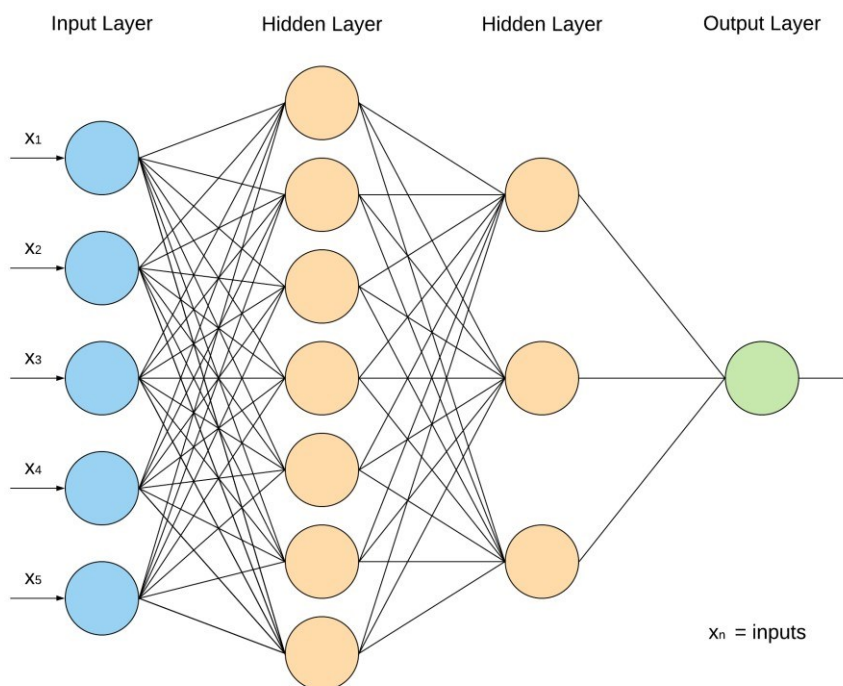


Figura 3 Schematic view of an artificial neural network (Khademi & Jamal, 2016, p. 3; Sayadi, Monjezi, Talebi, & Khandelwal, 2013, p. 320)

Artificial neural networks try to find a weighting matrix with learning rules, in which the desired result is realized with a tolerable error. Neural networks learn by modifying the connection weights in the frame of the repeated presentation of different sample patterns. The goal of the training process is to successively minimize errors between the expected output and actual output. According to Traeger et al. (2003, p. 1057), Zell (1996), the mean square error is usually used as a measure for the total error of the neuron matrix. (Traeger et al., 2003, p. 1057)

Training neural networks is not always without problems. If the generalization error is checked during training with an independent test set of training data, it will reach a minimum sooner or later. While the error for the training quantity decreases, the error for the independent cases usually increases. This effect is called overfitting and occurs when a too small training data set is learned too precisely. The result works so well during the training because the individual patterns are learned exactly. However, if unknown data is used then, only unqualified results are obtained. This is due to the fact that the artificial neural network did not generalize, but only remembered the previous patterns. The test data set must, therefore, be large enough. Experience shows that at least ten times as many training examples as connections in the network should be available. To identify overfitting, it is recommended that the neural network is regularly checked with the validation data set during training. According to Traeger et al. (2003, p. 1059), Zell (1996), this allows training to be completed with optimal results. In addition to the problem of overfitting, there is also the problem of an overlong training period. Traeger et al. (2003, p. 1059), Lawrence (1992) also mentioned that, when neurons are added, the training duration increases in the third power. For this reason, the number of input variables should be reduced to a reasonable level by preselecting them. (Traeger et al., 2003, pp. 1058– 1059)

## Natural Language Processing

Natural Language Processing (NLP) is a subfield of AI and a computer-based approach which deals with text analysis based on a number of theories and technologies. This includes all techniques that enable computer systems to interpret and produce natural language in word and writing. Human speech input is processed by algorithms and converted into machine-like representations. This includes the recognition of spoken language,

automated writing of texts in highly formalized areas such as financial or sports news, analysis of sound quality and mood in texts, conducting conversations and dialogues as well as machine translation. The great challenge here is semantic interpretation. (Döbel et al., 2018, p. 13; Liddy, 2001, p. 1; Rao & Verweij, 2017, p. 26)

Due to the large amount of text data, NLP is gaining in importance in today's world. With NLP, machines can analyze more language-based data than people do, in a consistent way and without maturity. Given the huge amount of unstructured data which is generated every day, from social media to medical records, automation is critical to an efficient analysis of text and speech data. NLP also serves to structure this highly unstructured data source, the complex and diverse human language. Supervised and unsupervised learning are widely used today to model human language. However, there is also a need for semantic and syntactic understanding and expertise. This is not available in these approaches of machine learning. NLP is important for this reason because it helps to eliminate ambiguities in the language. It also gives the data a useful numerical structure for many downstream applications such as text analysis or speech recognition. (SAS Institute Inc., 2019)

NLP involves a lot of different techniques for interpreting human language. These range from machine and statistical learning methods to algorithmic and rule-based approaches. A broad range of approaches is required. This is due to the enormous differences in the text-based and language-based data, as well as differences in the practical applications. The basic tasks of NLP include Part-of-Speech (POS) tagging, lemmatization/stemming, tokenization and parsing, identification of semantic relationships and language detection. NLP tasks break language down into shorter, elementary parts, try to understand the relationship between them, and investigate the interaction of the parts to create meaning. In higher-level NLP functions, underlying tasks such as machine translation, content categorization, document summarization, contextual extraction or sentiment analysis are used. The overriding goal is to take over the raw language input and to transform or enrich the text with the help of linguistics and algorithms in such a way that it delivers a higher value. To give an example of how NLP works, four important techniques are explained in more detail. For the example, the sentence "John hit the can" is considered. (SAS Institute Inc., 2019; van Duin & Bakhshi, 2018, p. 14)

The lexical analysis is one of the first steps of NLP. To do this, a technique known as POS tagging is used. When using POS, each word gets a tag that corresponds to a category of words with grammatical similarities. However, not only words but also whole sentences and paragraphs are tagged. POS is primarily performed with statistical models. These statistical models provide probabilistic results instead of hard if-then rules, which is why they are better able to process unknown text. In addition, they can handle the possibility of getting several possible answers instead of just one. (van Duin & Bakhshi, 2018, p. 14)

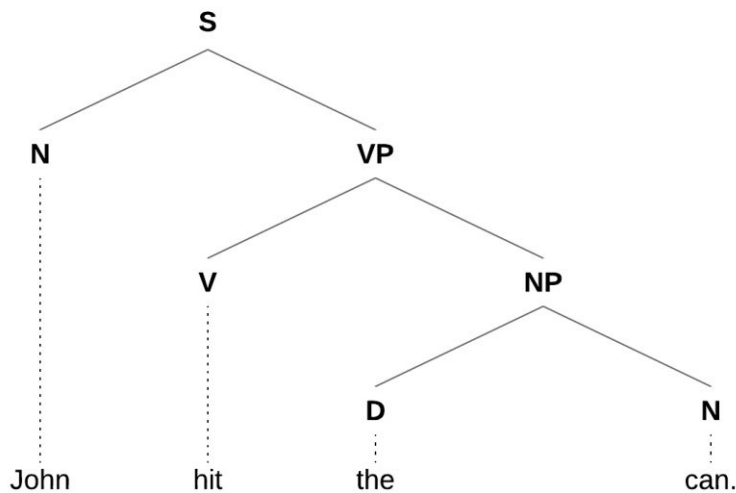
For tagging, a technique known as Hidden Markov Model (HMM) is often used. According to van Duin and Bakhshi (2018, p. 15), “an HMM is similar to the Markov Decision Process, where each state is a part of speech and the outcome of the process is the words of the sentence”. HMM’s are able to remember word sequences that have occurred before. This helps the HMM to estimate which POS a word is. Thus, for example, the word “can” in “the can” is classified as a noun rather than a verb. The final result then looks like this: “John” is tagged as noun (N), “hit” is tagged as verb (V), “the” is tagged as determiner (D) and “can” is tagged as noun (N). (van Duin & Bakhshi, 2018, pp. 14–15)

Similar to the POS tagging is also the technique Named Entity Recognition (NER). With NER, however, the words are tagged with the entity’s type that the word represents and not with the function they have. Examples of these entities are place, time, person, and company. Although NER can also use an HMM, the preferred technique is a Recurrent Neural Network, which is a different type of neural network. This neural network takes a number of words in a sentence or complete sentences as input. In addition, it remembers the previous sentence’s output. In the example sentence mentioned, John is recognized by the Recurrent Neural Network as the entity “person”. (van Duin & Bakhshi, 2018, p. 15)

In the technique called Parsing, which is a syntactic analysis, the text is analyzed for the arrangement of words and grammar. From this way, the relationship of the words is clarified. According to van Duin and Bakhshi, this is achieved by using the POS tag from “the lexical analysis and then grouping these into small phrases, which in turn can also be combined with other phrases or words to make a slightly longer phrase” (van Duin & Bakhshi, 2018, p. 15). This is repeated until the goal of using each word in the sentence is achieved. The words can be grouped according to the rule



called the grammar. The rule can have a form such as this:  $D + N = NP$ . In full, it means: Determiner + Noun = Noun Phrase. The result is depicted in figure 4. (van Duin & Bakhshi, 2018, p. 15)



*Figura 4 Constituency structure after parsing a sentence (van Duin & Bakhshi, 2018, p. 15)*

The S in figure 4 stands for sentence and forms the start symbol. It is a nonterminal element and the root node of all, by the grammar described, legal trees. (Bird, Klein, & Loper, 2015; Elhadad, 2012; Gauthier, 2016)

## Natural Language Understanding

Natural Language Understanding (NLU), a subsection of NLP that deals with machine reading comprehension, has gained popularity due to its potential in AI and cognitive applications. NLU goes beyond language's structural understanding to interpret and classify intention, resolve word and context ambiguity, and create a well-formed human language for itself. NLU algorithms address the complex problem of semantic interpretation. This includes understanding the intended meaning of written or spoken language with all the conclusions, contexts and subtleties that people can understand. Figure 5 below shows the demarcation between NLP and NLU and their techniques. (Ovchinnikova, 2012, pp. 15–16; SAS Institute Inc., 2019)

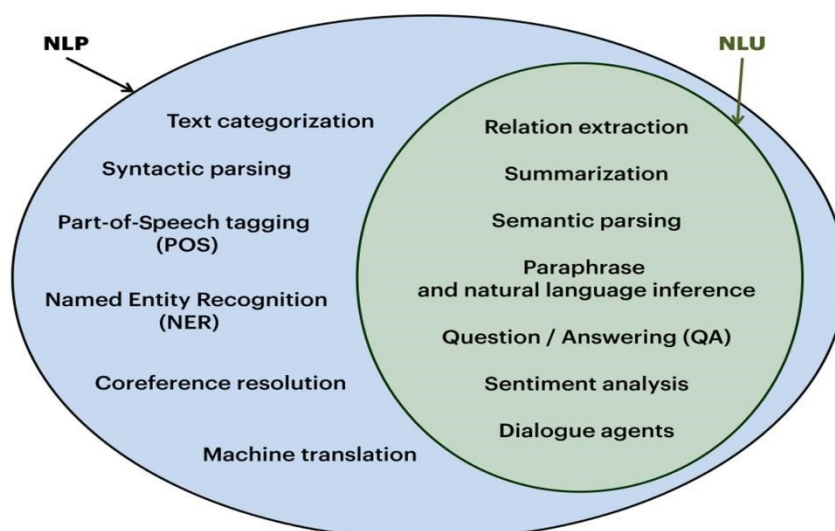


Figura 5 Demarcation between NLP and NLU (Olivier, 2017)

A quote about NLU mentioned by Raj (2019, p. 30), but not further specified, is as follows: “A good rule of thumb is to use the term NLU to express a machine’s ability to understand the natural language in a form provided by humans”.

With the emergence of NLU, conversational interfaces, especially chatbots as a very popular form of it, have attracted a lot of attention lately. Thanks to toolkits and NLU services, chatbots can now be created without much programming knowledge. (D. Braun, Hernandez-Mendez, Matthes, & Langen, 2017)

## Chatbots

Chatbots, chatterbots, Q&A agents or even Lingubots are software agents that offer access to a knowledge base through communication via an interface in natural language and an example of the application of AI to language. They are therefore computer programs that interact with users in natural language. Chatbots can be partly personified, which means that they can be displayed on the screen as humans, mythical creatures or animals. However, they can also appear as pure text input boxes. (A. Braun, 2003, p. 21; Geeb, 2007, p. 53; Shawar & Atwell, 2007, p. 29; van Duin & Bakhshi, 2018, p. 20)

The chatbot technology began in the 1960s with the goal of seeing if such systems could deceive users into thinking they were real people. The

purpose of today's chatbot systems is still the simulation of a human conversation, but with the aim to achieve economic advantages for companies and to increase the quality of service for customers. The following subchapters give an introduction to the topic of chatbots. (A. Braun, 2003, pp. 28–44, 97–109; Shawar & Atwell, 2007, p. 29)

### History of chatbots

Inspired by the Turing Test (1950), engineers and researchers have developed several conversation systems for chitchat. These first chatbots are computer programs that conduct conversations with text or audio. Such programs were often developed to simulate convincingly how people would behave as conversation partners. (Shieber, 1994, pp. 70–78; Shum, He, & Li, 2018, p. n.d.)

ELIZA is considered one of the first publicly known chatbots and was developed by Joseph Weizenbaum in 1966 at the Massachusetts Institute of Technology. Based on handmade scripts, it can communicate with people. These scripts, which simulate a Rogerian psychotherapist, can only accept textual input. It is not able to understand a conversation, it searches for suitable answers by means of pattern matching, combined with some intelligent phrasing. Due to limited knowledge, ELIZA can only chat with people within a restricted domain. Despite the limited range of knowledge, many users believed they were chatting with a real person when ELIZA first came out. (Shum et al., 2018, p. n.d.; Weizenbaum & Joseph, 1966, pp. 36–38, 41–42)

Parry was a chatbot developed by Kenneth Mark Colby (1975) in the early 1970s, who passed the first Turing Test by imitating the behavior of a paranoid individual. Although Parry is more advanced than ELIZA, it has a similar structure and is also rule-based. Compared to ELIZA, Parry had a better controlling structure, better language comprehension, and a better mental model. This mental model could simulate the emotions of the bot. For example, when the anger level is high, Parry reacts with hostility. (Chatbots.org, n.d.; Shum et al., 2018, n.d.)

Artificial Linguistic Internet Computer Entity also known as ALICE, was developed by Richard Wallace (2009). It is the first Artificial Intelligence Markup Language (AIML) based personality program that won the Loebner Prize in the category “the most human computer”. This prize is awarded at

the annual Turing Test competitions. ALICE won it in 2000, 2001 and 2004. AIML is a derivate of Extensible Markup Language (XML). The tags available in AIML allow bots to recursively call a Pattern Matcher to simplify the language. However, due to the constraints of AIML, the capacity of these AIML-based chatbots is limited. A reason why ALICE did not pass the ultimate Turing Test was that chitchat systems created with AIML cannot maintain the dialog over a long period of time. (Shum et al., 2018, n.d.)

### Current state of chatbots

In 2011, there was a breakthrough when IBM Watson beat people in a well-known American quiz show in a question-and-answer game. Recently came another breakthrough achieved by Google. They managed to give a chatbot a short-term memory that allowed the chatbot to mimic real conversations more realistically. (Döbel et al., 2018, p. 9; van Duin & Bakhshi, 2018, p. 20)

According to various reports, conversation interfaces like chatbots have become increasingly important in recent years. This is due to various reasons. One of these reasons is the improvement of chatbot's capabilities. Another one is the expectation for companies of customers which have changed due to the emergence and establishment of chat platforms in everyday life. It is therefore partly expected that companies will also move with the time and are attainable over written channels such as Messenger or chat. Figure 6 shows the chatbot trend over the last five years based on the web search term. (Klopfenstein et al., 2017, pp. 555, 557; Schulte, Zimmermann, & Müller, 2018, p. 395)



Figura 6 Chatbot trends of the last five years based on web search term (Raj, 2019, p. 2)

Braun et al. (2017, p. 174) think that the rise of universal chat platforms such as Slack, Telegram or Facebook Messenger, the progress in machine learning, and NLU as a service are the reasons for the recent hype around chatbots.

Looking at today's state of affairs, the explosion of chatbots could indicate that they all have been successful and that there are agents that can interact like humans. But the use of general chatbots is proving disappointing because they don't do what a user expects them to do. The narrow chatbots, however, do very well. The reason for the disappointing chatbot is the user's hyped expectations of its functionality. Chatbots are expected to be able to answer anything without understanding that the chatbot's job is to answer only relevant questions. Even people would not be able to answer questions that are not within their area of knowledge.

The goal is to reach AGI. However, this does not mean that every chatbot that is created fulfills this goal. As long as the chatbot does its task, it should be satisfying. (Nimavat & Champaneria, 2017, p. 1022)

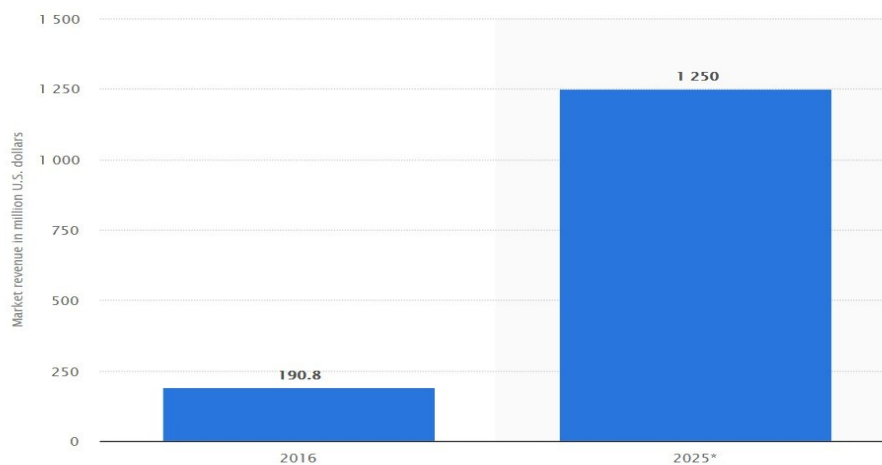
Bots are therefore not yet as understanding and accurate as humans. But they are able to do basic work. If they get stuck in an activity, a human can still take control. These are reasons why chatbots are seen as a domain with great potential for customer acquisition and promotion. (Jain & Jain, 2017, pp. 277–280; Nimavat & Champaneria, 2017, p. 1019)

### Future of chatbots

Shawar and Atwell have investigated various chatbot systems (2007, pp. 35–45) and believe that they will not be suitable for every area in the future. While some chatbots have performed well in areas such as entertainment and e-commerce, others have performed poorly in the area of education. They believe that “the teacher is the backbone in the teaching process” and that technologies such as multimedia presentations, chatbots or computer algebra systems can serve as amplifiers, but cannot replace good guides (Shawar & Atwell, 2007, p. 45). Shawar and Atwell also mentioned that the overall goal of chatbot designers should be to develop tools that help people to make their work and interaction with natural language computers easier. It would be wrong, in their view, to completely replace the human

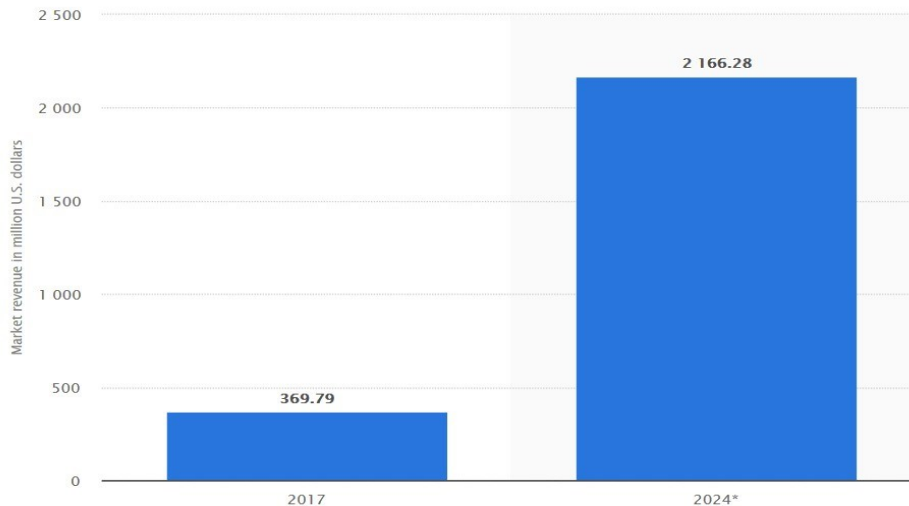
role or perfectly imitate human conversations. (Shawar & Atwell, 2007, pp. 35–45)

Based on Statista’s statistics (2017b, 2018), the future of chatbots looks positive. According to one of their statistics (2017b), the global chatbot market will be more than 6.5 times bigger in 2025 than it was in 2016. Figure 7 shows the result of the survey carried out by Statista on the revenue generated of the global chatbot market. In 2016, this was valued at 190.8 million US dollars worldwide. (Statista GmbH, 2017b)



*Figura 7 Size of the chatbot market worldwide, in 2016 and 2025 (Statista GmbH, 2017b)*

One year later, the market was reassessed and, based on this, a further statistic was published, in which it is evident that the size of the chatbot market will be even larger than had been assumed a year earlier. The results of Statista’s survey in Figure 8 show the global chatbot market in 2017 compared to the forecast for market size in 2024. The global chatbot market was valued at nearly 370 million US dollars in 2017. Statista’s forecasts indicate that it will increase significantly in the next few years. (Statista GmbH, 2018)



*Figura 8 Size of the chatbot market worldwide, in 2017 and 2024 (Statista GmbH, 2018)*

If Statista's forecasts can be trusted, chatbots will, therefore, become more attractive for many in the near future. According to Klopfenstein et al. (2017, p. 564), chatbots will not replace the entire ecosystem of mobile applications in the near future. Nevertheless, special attention must be paid to the rise of bots as new software platforms that can provide services and data to users.

### Need for chatbots and their benefits for companies and users

According to Raj (2019, p. 5), it is time for companies to treat chatbots as a marketing tool. The reasons given for this are accessibility, efficiency, availability, scalability, cost, and insights. A good argument for companies to use chatbots is that they generate revenue. As stated by Raj (2019, p. 6), companies that start with chatbot support or create a new chatbot to support customer requests are doing well on the market compared to their competition. For example, two months after the launch of its Facebook chatbot, a US floral and gourmet food gift retailer reported that more than 70% of orders via its Messenger were new customers, which ultimately contributed to a significant increase in annual revenue. It can also be added that chatbots, using the latest machine learning and data science techniques, can give companies an exclusive insight into the behavior patterns of their consumers. In addition to making money, the use of

chatbots can also save costs. Here the scalability can be mentioned, for example. If a bot can do what a customer needs, it can handle hundreds of thousands of customer requests simultaneously and handle the same or new tasks with the same efficiency and performance every time. This means that employees can be used more efficiently. In addition, customers no longer have to wait until a customer representative becomes available. (Raj, 2019, pp. 5–10)

Considering the customer aspect, it can be said that customers as users prefer chatbots to other software systems for human-computer interactions. On the one hand, they are easily accessible and can be used immediately by the customer. On the other hand, they are available 24 hours a day, seven days a week and can be used by customers from anywhere. In addition, it is not necessary to learn a formalized language in order to interact with a chatbot. Users can use the natural language within an interface that requires minimal learning effort. Bots do not have to be downloaded or installed but can be used immediately. Therefore no hard disk space is used, and no complex configuration by the user is required. New applications normally require registration and account creation to allow interaction with the service, while chatbots do not require user authentication. This saves chatbot users the long registration process of creating a user account or forgetting access credentials for the service. By default, the chatbot uniquely identifies users, so no additional passwords or accounts need to be created. (A. Braun, 2003, p. 28; Klopfenstein et al., 2017, pp. 561–562; Raj, 2019, pp. 5–10)

### Terminologies used

According to the founder of Tangowork Software Chris McGrath (2017), the three most important words when it comes to chatbots are intent, utterance, and entity. The user's intention is called intent. For example, a user's intention might be to retrieve a list of financial messages when the sentence "show me yesterday's financial news" is sent. The names of intents often consist of a noun and a verb. Utterances are different forms of the same intention or question that a user can have. But utterances are used on the other side as well, by the chatbot itself in the form of answers. Intents can contain metadata that can modify an intention. These are called entities. In the example of listing the financial news, "yesterday" and "financial" are entities, because it could have been "today" instead of



“yesterday” and “sports” instead of “financial”. General entities can be used so that they can be used for all intents. Entities can be represented as a number, volume or quantity. Multiple entities can be used within one intent. Figure 9 is intended to illustrate the terms described above visually using the example sentence with the financial news. (McGrath, 2017; Raj, 2019, p. 27; Treml, 2018c)



Figura 9 Illustration of the three most important terms using a sentence (McGrath, 2017)

Another term is the confidence score or confidence level. After each query of an intent classification for an utterance a confidence score is given, which states how confident the machine learning model just used is, when recognizing the intention. The confidence score can usually be improved by increasing the training data. The more utterances there are, the better the model and the higher the confidence level. A too high confidence level could be an indicator for overfitting. It is important to mention that the confidence level is not a meaningful statement of the probability that the prediction is correct. It is merely a model-defined metric that broadly describes how closely the input was similar to the training data. Table 1 lists examples of the confidence level of Microsoft's cloud-based Language Understanding Intelligent Service (LUIS). (Bansal, 2018; Microsoft Corporation, 2019; Raj, 2019, p. 28; Rasa Technologies GmbH, 2018d)

Table 1 Microsoft's confidence level examples for the cloud-based solution LUIS (Microsoft Corporation, 2018)

Score value	Confidence
1	definite match
0.99	high confidence
0.01	low confidence
0	definite failure to match

### Types of chatbots

Chatbots can be classified into different types based on the knowledge they have or which they access, their level of interaction, their method of response generation or the goal they want to achieve. Figure 10 illustrates the classification of a chatbot according to the four parameters knowledge domain, service provided, goals and response generation method. (Nimavat & Champaneria, 2017, pp. 1019–1012)

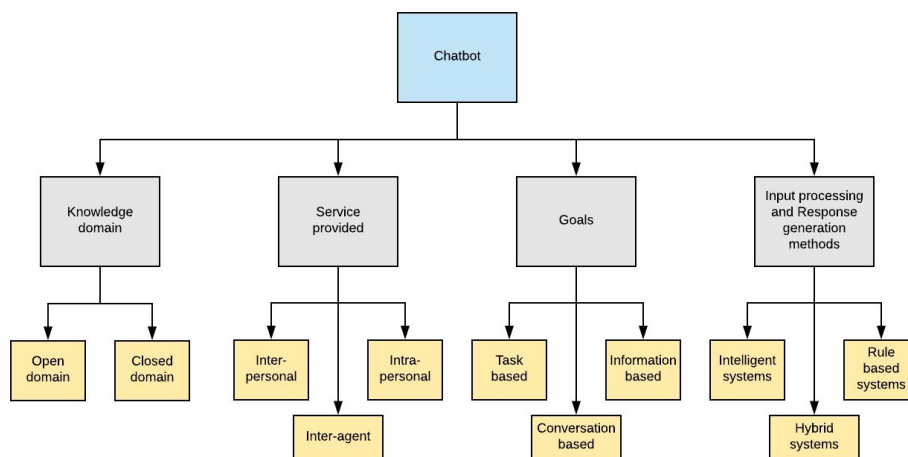


Figura 10 Classification of chatbots (Nimavat & Champaneria, 2017, p. 1020)

If the chatbots are classified based on the knowledge they have or they access, they are divided into the classes open domain and closed domain.

Open domain bots can talk with the user about general topics and can answer the questions appropriately. Examples of open domain bots are Amazon's Alexa or Apple's Siri. In contrast, to open domain bots, closed domain bots are focused on a specific task and are only trained for this specific knowledge area and are much more common. Therefore, such bots may not be able to answer questions that are not part of their knowledge domain. A restaurant booking bot would be an example of a closed domain bot. This chatbot will therefore only be able to perform tasks such as table reservations or providing information about the restaurant. Questions that do not affect the knowledge area of the bot may be answered in the form of jokes or his mood. (Britz, 2016; Nimavat & Champaneria, 2017, p. 1019)

If service provided is regarded as a classification parameter, chatbots have characteristics of one of the three types interpersonal, intrapersonal or inter-agent. In this case, the bots are classified based on proxemics. This does not mean the physical proximity to the user, but the sentimental proximity, the amount of intimate interaction with the user, and therefore the type of task the bot performs. Interpersonal chatbots are located in the proxemics chart at the social or personal distance range when it comes to communication. Examples of such chatbots are bots that offer services such as restaurant or flight bookings. These types of chatbots are not intended to be companions to the user but are used solely to receive and share information. Although they may have a personality, be friendly and remember the user, they are not obliged to do so. Intrapersonal chatbots are bots that exist within the user's personal domain and therefore perform tasks from the user's personal domain. Examples of such bots are chat apps such as Slack, Messenger, and WhatsApp. In contrast to the interpersonal bot, the intrapersonal bot is regarded as a companion of the user and understands him as a human does. These may not be dominant at the moment, but due to the improvement of NLU, they will soon prevail. When talking about inter-agent bots, it means two systems communicating with each other to perform a task, which is why it is reasonable that this type of bots will be widely used in IoT dominant areas. Since bots are becoming ubiquitous, any bots will need inter-bot communication capabilities, which will drive the need for protocols for inter-bot communication. An example of inter-agent communication is the integration of Amazon's Alexa and Microsoft's Cortana. (Nimavat & Champaneria, 2017, p. 1020)

When it comes to the primary goal that a chatbot should fulfill, the chatbots can be classified into the three types information based, task based and

conversation based. Information based bots provide users with information that comes from a fixed source or was previously stored. They are usually information retrieval algorithm based and would either perform string matching or retrieve results from database queries. In most cases, the information sources are static such as frequently asked questions (FAQ) pages or warehouse databases with inventory entry. An example of information based bots are FAQ bots. Task based bots perform certain tasks, such as assisting in browsing a store or booking a flight. In most cases, the actions that contribute to the fulfillment of a task are predefined and the flow of events is defined as well. Examples of task based bots are restaurant booking bots. Conversation based bots talk to the user as people do to each other. Their goals are to respond correctly to the sentences they receive from the user and to continue the conversation with the user using techniques such as cross-questioning, deference and evasion. Examples of conversation based bots are Siri, Alexa, Jenny, Tay or Xaoice. (Neff & Nagy, 2016, pp. 1–2; Nimavat & Champaneria, 2017, p. 1020)

When classified according to the input processing and response generation method, chatbots can reflect one of three types of intelligent systems, rule-based systems, and hybrid systems. The intelligent systems can generate answers and use NLU to understand the queries. These types of systems are used for narrow domains and when sufficient data is available for system training. Rule-based chatbot systems use pattern matching and can be used when the number of possible results is specified. These types of systems have proven to be rigid. Hybrid systems are a mix of machine learning and rules. An example of a hybrid system would be a bot that manages the conversation's directions using a flowchart but provides answers that are generated using NLP. (Nimavat & Champaneria, 2017, p. 1020)

### Processing of a chatbot

According to Bihade, Badhiye and Shelke, the functionality of chatbots is divided into the two areas “getting the intent of user message” and “generating the response”. The processing of messages begins with understanding the things the user is talking about. The intent classification module identifies the intended message of the user. Usually, it is a selection

from a set of predefined intents. More complex bots are able to identify multiple intents from only one message. The intent classification can use contextual information such as settings, user profiles, or previous message intentions, as needed. Structured bits of information can be extracted from the messages using an entity recognition module. For example, a weather bot could extract date and location from a message. (Bihade, Badhiye, & Shelke, n.d., p. 64)

A candidate response generator performs all domain-specific calculations to handle the user request. It can call external Application Programming Interfaces (API), use different algorithms, or ask a human to help with the generation of the response. The output of these calculations is a set of responses that should be correct according to domain-specific logic and not just a collection of random responses. To support multi-message conversations, the response generator must use the entities, intent, and context of the conversation from the last message of the user. A response selector then evaluates all possible responses and returns the one that is most suitable for the user. (Bihade et al., n.d., p. 64)

### Training and testing

When talking about training chatbots, the term essentially refers to creating a model that learns how to categorize new utterances from an existing set of defined intents, entities and utterances. The categorization is measured on the basis of a confidence score. (Raj, 2019, p. 28)

According to Florian Treml, founder and chief executive officer of Botion GmbH (2019), a company dedicated to the training, testing and quality assurance of chatbots, “a well-designed chatbot is trained to recognize most common utterances”, which means that for each step in the conversational flow chart of a chatbot, various output and input utterances have to be considered. He mentioned that the coverage of statements for an agreement such as “yes” or a negation such as “no” with 12 different statements each may suffice, but is far from complete status. (Treml, 2018c)

Treml mentioned that most chatbots are of poor quality because they either have little or poor training data or omit or neglect the training of the chatbot. “The difficulty and high effort comes from implementing a process

for training the bot, and that's where lots of companies are failing" (Trembl, 2018a). Therefore, it can be said that chatbots are only as good as

the training they receive. The training quality is in turn only as good as the training data it contains. The best data quality is ensured if it is determined in conversations with real customers and users. The price for this is a higher effort in collecting and evaluating the training data, which could take several months of effort until an appropriate amount has been collected. In addition, there is the training effort, which is very high from a certain number of training data, and never ends. Trembl noted that the key to success lies in constantly training and monitoring the chatbot to make it smarter. This can be achieved either by planning regular training cycles that include new utterances and conversations from real users or by continuous training with human effort. (Trembl, 2018a)

Attention should be paid to overfitting. Otherwise, the created model does not generalize well from training data to new, unseen data. To detect overfitting, it is important to note that the training data is different from the test data. This is the only way to ensure that the model to be tested really delivers good results. Separating the training data from the test data, however, does not prevent the problem of overfitting. It is for identification purposes only. Overfitting can be avoided by various methods. Among others with the help of cross-validation, training with more data, early stop of the training process, regularization, which means to simplify the model, removing features or by ensembling multiple separate models. (EliteDataScience.com, 2019)

Testing a chatbot can cover several areas. An essential part is the evaluation of the conversation to find out or evaluate how satisfactory or unsatisfactory the conversation between the user and the chatbot is (Chakrabarti & Luger, 2013, pp. 34–35; Russell-Rose, 2017).

To answer these, metrics must be used appropriately. In a commercial environment, these can be key performance indicators (KPI) and business figures, but not all chatbots are used in such an environment. Additionally, the question arises how these metrics as part of a feedback loop can lead to iterative improvements. For example, different perspectives of the chatbot development can be considered in this case. The four perspectives information retrieval, user experience, linguistic perspective, and AI perspective are mentioned. Russell-Rose (2017) believes that while the information retrieval perspective provides the most basic approach to

quantitative valuation and the user experience perspective provides the qualitative valuation, none of these perspectives can provide a universal framework and a hybrid valuation scheme may be required instead. Since not all chatbots are the same, it is possible to narrow down the problem space and consider the dimensions in which they move. However, no final answer will be provided. Instead, it gives a basis that allows thinking about metrics and rating frameworks. (Russell-Rose, 2017)

Muñoz-Romero, Ábalos and Martín (n.d., 2017) are of the opinion that for the evaluation of a chatbot, questions of the following seven categories should be considered, in order to test the specific functionalities of a chatbot and improve them accordingly: (1) Personality, (2) Onboarding, (3) Answering, (4) Understanding, (5) Navigation, (6) Error management, (7) Intelligence. Table 2 shows the superordinate question of the respective category.

Table 2 The seven categories of chatbot's design (Muñoz-Romero et al., n.d., 2017)

Category	Explanation
Personality	Does the chatbot have a clear voice and tone that fits with the users and with the ongoing conversation?
Onboarding	Do the users understand what the chatbot is about and how to interact with it from the very beginning?
Answering	What elements does the chatbot send and how well it is doing it? Are they relevant to the moment and context?
Understanding	Requests, smalltalk, idioms and emojis. What is the chatbot able to understand?
Navigation	How easy is it to go through the chatbot conversation? Does the user feel lost while speaking with the chatbot?
Error management	How good is the chatbot dealing with all the errors that are going to happen? Is it able to recover from them?
	Does the chatbot have any intelligence? Is it Intelligence able to remember things? Does it use and manage context as a person?

According to Treml (2018b), the difficulty of testing a chatbot has several reasons. One of them is dealing with non-deterministic user interactions when testing software. Due to the human language's nature, it is not possible to cover all possible situations in software tests, as a single sentence can take many forms with the same meaning. Therefore one should not think of a 100% test-coverage. It should be ensured that the most common situations are covered in the test. Another reason is that there are no barriers for users. In the middle of the conversation, users can change topics or the direction of the conversation with which the developer does not expect and which ultimately cannot be handled by the chatbot.



For this reason, test cases should be developed that are robust for unexpected user input. (Trembl, 2018b)

## METHODS

This chapter explains the methodological procedure used for data collection, development and implementation of the chatbot. All results of the methods used have an influence on the practical part of this thesis.

### Project management

Since the development of a chatbot in this research scenario is a unique and complex activity and has a fixed start and end time, the practical part of this thesis can be regarded as a project. Projects should be managed according to a certain method or a certain process model because they show the project tasks in a process-oriented way and give the project members instructions for action. Although projects can be successfully implemented by applying non-transparent or self-created procedures, in reality they usually fail, which is why setting up a project without a method is considered negligent. If a project is started without a method, then secretly an own process is developed, with phases, stages, milestones, and tasks. Such behavior costs time and money, so these projects run out of time and schedule. The use of a method that is not used for its own purpose, therefore, makes a major contribution to ensuring the success of the project. (Feyhl, 2004, p. 14; Jenny, 2014, pp. 38–39, 56–57)

Although there are no monetary goals to be achieved and the project organization consists of very few people, an agile process will be used for the project part of this thesis. Due to the size of the project and the application area, however, a tailor-made process, known as process tailoring, is applied, which is customized to the needs of the project situation and therefore optimized (Pröpper, 2012, pp. 132–133).

## Process model

Due to possible, dynamic adjustments to the system, the need to ensure the development of creativity in the application of the training data and the possibility of changing general conditions, an agile method is used for the practical part of this thesis. Kanban is a process model, which can be used in agile software development, for reducing ballast, called waste that impedes the progress of work. The recorded and prioritized requirements are visualized on a Kanban board, which is then processed in the order of their priority and which can be defined according to criteria such as importance, urgency or risk. (Epping, 2011, pp. 1, 27, 73, 115; Kraus, 2009, p. 6; Wang, Conboy, & Cawley, 2012, pp. 1287–1289)

## Meetings

Weekly meetings, in which information is exchanged, existing problems discussed, questions clarified and the progress of the project reviewed, are intended to ensure that nothing stands in the way of the development and success of the project (Badertscher, Geiger, Gubelmann, Pifko, & Romano, 2014, pp. 111–112, 199–202; Züger, 2013, pp. 124–125).

## Project boundaries and system delimitation

Projects have a certain scope which should be discussed at an early stage to prevent disappointments. For this reason, it is advisable to work out project boundaries and system delimitations to clearly show which work is done and which is not. The following subchapters describe the boundaries and delimitations specified with the responsible of The AI Institute.

## Prototype

The results of this research are based on a self-developed chatbot prototype that is integrated into a self-created website. The future maintenance of the data, the further development, use and operation of the chatbot will not be part of the thesis.

## Target group

The chatbot will only support the English language. The support of other languages is not part of this thesis and will therefore not be covered. This is why an English language model is used for the NLU when creating the chatbot. The target group of the chatbot is made up of exchange students, international students, double degree students and locals who want to start their studies at The AI Institute.

## Data collection

Although the training data are a central part of this research work and, according to Rasa Technologies GmbH (2018h), the crucial component in the development of a chatbot, it is primarily a matter of investigating the suitability of the open source product at the AI Institute as a whole. For this reason, the numerous data collection is not in the foreground, which is why a large amount of data sets are not collected. Data collection has, therefore, affected a small number of people. Nevertheless, an attempt were made to ensure a collection of qualitative data by asking the right people. The aim was to use creativity techniques to collect ideas from at least four exchange students, to obtain at least 25 results from an online survey and to conduct an interview with at least one person from The AI Institute.

## Scope of data

Since conversations between two participants can be an endless story, only a subset of possible questions and answers will be considered when evaluating training data. Thus a maximum of 50 possible chatbot answers is defined, which can be covered in the implementation.

## Data collection

Since the product to be developed in the practical part of this thesis is a machine learning based chatbot, data is needed to train it. In order to obtain realistic data, the methods interview, brainstorming and survey are applied to people who are connected to The AI Institute. (Döbel et al., 2018, p. 8; Rasa Technologies GmbH, 2018n)

## Survey

The purpose of the survey, which was completed by international students is to identify the questions that the students have dealt with. Since in the practical part of this thesis an NLP-based chatbot will be developed, training data is indispensable (Rasa Technologies GmbH, 2018h). The information obtained through the survey, together with the information from the other two methods, brainstorming/brainwriting and interview, should serve as a basis for determining the training data.

The structure of the survey is based on the theory and the aspects described in the book “Umfrage – Einführung in die Methoden der Umfrageforschung” (Jacob, Heinz, & Décieux, 2013). Examples of the aspects considered are: (1) What is the central question that should be answered, (2) Is the question formulated as a complete sentence, (3) Can we assume that the respondents have the necessary knowledge to answer the questions. It should be noted that in the preparation of the survey, questions were formulated which refer to the “everyday knowledge” of the interviewees, which is seen as a function of individual experiences, evaluations and expectations, and not to the so-called “expert knowledge” (Jacob et al., 2013, pp. 42–43).

## Brainstorming and brainwriting

The brainstorming and brainwriting methods are used to gather information for the training data of the chatbot as well. In contrast to the survey, which collects the information that students already have in their minds, brainstorming, and brainwriting, as intuitive-creative methods, help identify by collecting ideas potential new questions that have not yet been asked to, but still come into question, or even have arisen in the past from other students (Kollmann, Kuckertz, & Stöckmann, 2016; VanGundy, 1984).

For the application of this methods the theories from the books “Praxishandbuch berufliche Schlüsselkompetenzen” and “Das 1 x 1 des Wissenschaftlichen Arbeitens”, from the article “Brainwriting for New Product Ideas: An Alternative to Brainstorming” and from the web article “Brainwriting... this is a team exercise to develop new and creative ideas.” have been used (Becker, 2018; Kollmann et al., 2016; Robinson, n.d.; VanGundy, 1984). At this point, it is worth mentioning that instead of the

usual 635 method of brainwriting, in which six participants pass on three ideas five times each, the less formal brainwriting variant, the so-called brainwriting pool, has been used. The participants were informed about the subject area, where upon they independently updated a list of ideas until the flow of ideas is exhausted from their point of view. These lists are passed on until each participant has seen each list once and added his ideas. (Kollmann et al., 2016, p. 12)

## Interview

The implementation of a third method, the interview, is intended to identify effective questions from exchange students, as well as their number, frequency, and relevance. In order to obtain relevant data, the international student was interviewed. Due to the questions that have to be asked in a fixed order from a predefined list, the interview type used was a guideline-based interview, also known as a guided interview (Helfferich, 2014, p. 565).

For the application of this method, the theory from the books “Handbuch Methoden der empirischen Sozialforschung” and “Methoden in der naturwissenschaftsdidaktischen Forschung” is used (Helfferich, 2014; Niebert & Gropengießer, 2014). It should be mentioned that the thematic structure was taken into account in the preparation of the interview form. The evolution of the interview guide was also taken into account. (Niebert & Gropengießer, 2014, pp. 128, 130)

## Real conversation

In addition to the three methods for collecting the training data, on the recommendation of Rasa Technologies GmbH another method is applied, the testing of the chatbot by humans (Rasa Technologies GmbH, 2018i). Rasa Technologies GmbH believes (2018i) that real conversations between people and the AI are more important than hypothetical approaches to get good or even the best training data. For this reason, after the creation of the chatbot, the opinion of fellow students is obtained by testing the chatbot on themselves. The aim is to ensure that missing or unsuccessful communication between a person and the chatbot, which remains

undiscovered during the implementation of the above three methods, occurs, is improved and then included in the training data as a supplement.

### Consultation of experts

Since the manufacturer of the used software product is a young company and the author of this thesis is aware of a single book about the product, the knowledge and the experience of experts of this world is consulted for the realization. On platforms such as GitHub or forums such as the Rasa Community Forum, the knowledge of the experts was sought through interaction with them and asked as soon as the author had knowledge gaps or any problems.

## DATA COLLECTION AND TRAINING DATA

The following chapter contains the results of the data collection methods carried out and the created questions based on them, to which the chatbot should provide answers.

### Results

In the following subchapters, the information relevant to the training data of the chatbot from the survey, the creativity techniques used and the interview is mentioned. Demographic and personal data was not provided. No corrections or adjustments were made to the data. Only information classified as useful by the author was considered in this subchapter.

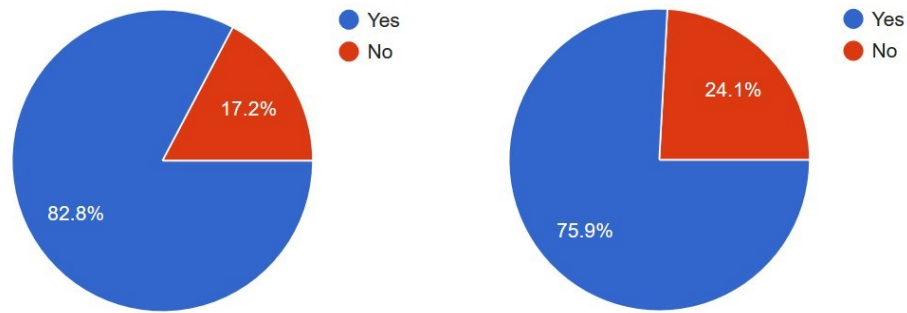
### Survey

During two weeks an online survey was made available to the exchange students present at University of Paris Saclay and the exchange students who spent their last semester at Paris Saclay in France. After 14 days, the survey was completed by 29 students, who are enrolled in nine different degree programmes. The survey questions can be found in appendix 1.

An insight, that can be used for the future is the acquisition of information by students regarding school-relevant issues.

More than 58% of respondents also said they use personal channels such as email or WhatsApp to get the necessary information from other exchange students.

One question dealt with the future use of the chatbot. The students were asked whether they would consult a chatbot if it was available on The AI Institute website. The question then asked, which was asked to the persons who would use a chatbot, is whether they would use the chatbot as the first port of call for questions. Figure 11 shows the results of these two questions. The left pie chart shows the results of whether a chatbot would be used, while the right pie chart refers to the first port of call question.



*Figura 11 Survey results: The use of a chatbot*

When asked whether a chatbot would be used by students to answer open questions if a company chatbot would exist, 24 of the respondents, i.e. more than 82%, stated that they would. However, five of the 29 students surveyed, which is about 17.2%, replied that they would not ask a chatbot for advice. Of the 24 students who would use a chatbot, 22 said they would use it as their first port of call in case of questions. That is more than threequarters of the students surveyed.

### Brainstorming and brainwriting

The creativity techniques brainstorming and brainwriting were conducted with certain students to obtain more data. Three students of the current semester and three students of the last semester were involved. The exchange students studying at Paris Saclay in the current semester were encouraged to share their ideas in a ten minute brainstorming session, as suggested by Kollmann, Kuckertz and Stöckmann (2016, p. 12), on possible and realistic questions that future students might ask to The AI Institute. The students of the last semester were given this opportunity to do so digitally. The latter were given three days.

The results obtained through brainstorming and brainwriting showed that the questions that could be asked by potential or future The AI Institute students can be divided into the five categories: university, administrative stuff, accommodation, transport possibilities and leisure time. The category university contains all questions related to the school, the degree programmes, modules or the infrastructure. Administrative stuff includes information about dentists in the region, mobile phone subscriptions, banks or the local registry office. The accommodation category includes any



questions about student accommodation such as keys, furnishings, price or location. The transport possibilities are understood as transport possibilities in France in general. The leisure time category contains questions about the activities a student can do during his free time. This includes sports, shopping or going out.

In addition to the results of the first session, further ideas were added, which can be divided into the two categories: small talk and ignorance. The small talk category included ideas that belong to a student's pastime. Examples of the mentioned ideas were questions about the weather, questions about the mood of the chatbot or general greetings in conversation. In the ignorance category, ideas came about how a chatbot should behave in the event of ignorance. Instead of an unnatural reaction or providing a wrong answer, if the chatbot does not know the answer to a question asked, it should respond with an appropriate answer. This could be in the form of a joke or offering appropriate contact information of a person who may know the answer.

## Interview

An interview was conducted by two of my colleagues (interview 14 May 2019), at The AI. However, since no meeting took place and the answers to the questions were only dealt with digitally by us, it is not considered to be an interview but is nevertheless classified in this chapter on the basis of its content and the questions asked, which corresponds to that of the interviewee.

The content of the requests varied, but all two were regularly confronted with requests for degree programmes. In addition to requests about degree programmes, questions were also asked about the location and campus, accommodation, the application process, eligibility requirements, entrance examinations, technical issues, tuition fees, and scholarships. According to the three, most of the information, which is requested, is also available on website of The AI Institute. They would welcome a chatbot that could answer frequently asked questions.

## Evaluation of training data

Based on the results of the applied methods, questions were evaluated that could be considered for the chatbot as training data. The evaluation has not been carried out according to any particular method but is based on the author's impressions and findings of the results. On the one hand, because this is not part of this research work, and on the other hand, because the prototype chatbot is not used productively and is therefore only presented to a small number of people. If The AI Institute decides that the prototype or at least its data model should be used productively for future missions, further training data can be entered and the existing data model can be extended. This would also be necessary since a chatbot of the planned size does not bring much use. Which training data are included exactly in the implementation does not matter as long as they are questions asked by students or answered by The AI Institute.

The analysis of the three data collection methods showed that the topics of university and accommodation, in particular, met with the greatest interest. However, there are also a number of requests for information on leisure activities or transport options. Based on these findings, the areas manners, accommodation, university, leisure time and administrative tasks were selected to be partially covered by the chatbot.

## Manners

Contact between parties, regardless of form, whether written, oral, live or digital, usually begins with a greeting and ends with a farewell. It is most likely that chatbot users will also be able to direct or end the conversation with the conversational agent in this way. The second brainstorming session also showed that this was also the view of the participants. However, they would like to do small talk and ask the chatbot for example about his mood in addition to greeting and saying goodbye. What also seemed important to them was that the chatbot did not simply remain silent in case of ignorance, or even provide a wrong answer, but skillfully and honestly said that this exceeded his area of knowledge. An almost perfect solution would be to forward the request to the responsible person or to offer a link leading to the possible solution.

## Accommodation

As the findings from the survey, most students felt that the chatbot should have knowledge about the student apartments. Not only the students but also The AI Institute's employees interviewed and surveyed were of the opinion that information about the accommodations was an important point to be covered by the chatbot. For this reason, questions about the accommodation are part of the training data. This includes questions about the price, location or equipment available in the apartments. But also the responsibilities, the contact data and the procedure in case of problems are covered.

## University

According to the survey, but also due to the information of The AI Institute ' staff, the need for information about the university is available among the students. In addition to the information of the international students, the findings of the Admission Services, in particular, showed that many of the potential students do not deal with the information on the website or do not find it at all and thus make inquiries via e-mail. If a chatbot were available on the website to answer the most frequently asked questions, the number of e-mails with FAQs sent to the Admission Services and the international office could be reduced. Thus, questions concerning both international and normal students will be included in the training data. These include questions about the application process, tuition fees, application deadline.

## Leisure time

Based on the survey, many students found information about leisure activities such as sports, shopping and general information about the city important. Therefore, questions about leisure activities were also included in the training data. This includes questions about sports opportunities and about things that can be done in the city to combat boredom.

## Administrative tasks

Especially from the information of the students, it follows that information on administrative matters is necessary for them. For this reason, questions from this area will also be included in the training data. The training data thus includes questions about the local register office.

## TECHNOLOGIES AND TOOLS

The technologies and tools described in the following chapter will be used for the development of the chatbot prototype. The description is only superficial. Detailed information about the software used, the system environment and the individual installed packages can be found in the appendices 2 and 3.

### Linux

Linux is a free operating system, which corresponds to the open source basic idea, the system as well as all available programs are usable free of charge without restriction. Due to its transparency, stability, and security, Linux is one of the most popular and widely used operating systems for servers, routers, and mainframes. In contrast to commercial operating systems, Linux permanently offers its users the possibility to better understand the functioning of the operating system and its programs. (Grotz, 2017, p. 1)

The combination of specified software into one compiled and configured package is called distribution. There are many different Linux distributions that are better or less suitable depending on the area of application. The Linux distribution used for this research work is Ubuntu and was, as well as the operating system, given by the client. (Grotz, 2017, p. 5)

### Apache Hypertext Transfer Protocol Server

The Apache Hypertext Transfer Protocol (HTTP) Server is a project initiated by the Apache Software Foundation to develop, extend and provide a robust, functional, commercially viable and freely available source code implementation of an HTTP web server. Named the Apache HTTP Server Project, the collaborative software development effort is led by a group of volunteers around the world who use the web and the internet to plan, communicate and develop the server and its associated documentation. (The Apache Software Foundation, 2019)

## Python

Python is a programming language further developed and published by the Python Software Foundation, which can be regarded as both an object-oriented programming language and a scripting language. Python requires relatively few keywords, is characterized by its simplicity, clarity, and extensibility, and has a large number of scientific program libraries in the field of data science and is thus becoming more and more the central tool. (Huttunen, 2015, p. 4; Python Software Foundation, 2001; Selle, 2018, p. 54)

Since the NLU used is written by Rasa Technologies GmbH in Python, this programming language is also used for the development of the chatbot prototype. On the one hand there is a detailed installation guide and product documentation with Python examples on Rasa Technologies GmbH's website and on the other hand, the import of relevant classes is simpler than if this is handled via an HTTP API. (Rasa Technologies GmbH, 2016, 2018m)

## Anaconda Distribution

Anaconda Distribution is an open source data science distribution for the programming languages R and Python, developed by the manufacturer Anaconda Inc. (2019a). By using Anaconda, as one point access for the administration of all necessary packages, many of which are already preinstalled, tools, libraries, and their dependencies can be installed and managed faster and easier. In addition, it is possible to manage multiple Python versions using Anaconda. The management of multiple Python versions might be important or even essential if in the future there will be more applications running on the server parallel to the chatbot, which require a higher or lower Python version. (Anaconda Inc., 2019a, 2019b; Davis, 2016; Misra, n.d.; Mortensen & Zonca, n.d.)

Furthermore, the chatbot manufacturer Rasa Technologies GmbH recommends the use of Anaconda when using the Rasa Core component (Rasa Technologies GmbH, 2018b).

## Rasa Stack

The Rasa Stack is a collection of open source machine learning tools from Rasa Technologies GmbH that can be used to create contextual chatbots and AI assistants. It consists of the two independent components Rasa NLU and Rasa Core, which can be used in combination as well as separately. Rasa Technologies GmbH recommends using both. (Rasa Technologies GmbH, 2018n)

## Rasa NLU

Rasa NLU is an open source NLP tool for the extraction of entities and the classification of intents. By intent classification, one understands the interpretation of meaning on the basis of predefined intentions. Entity extraction is the recognition of structured data such as e-mail addresses or telephone numbers. With The Rasa NLU, user messages can be understood based on previous training data. (Rasa Technologies GmbH, 2018n, 2018m)

Based on the research of Braun, Mendez, and Matthes (2017), which examined and evaluated current English NLU services, the Rasa NLU scored well compared to its competitors LUIS, Watson Conversation, API.ai, wit.ai, and Amazon Lex and came in second. The products have been evaluated on the basic functionality, based on two different data corpora. Due to the adaptability of the open source product Rasa NLU, even better results could have been achieved after some adaptations. (D. Braun et al., 2017, p. 180) According to Raj (Raj, 2019, p. 106), “Rasa NLU is not just any other library with a bunch of methods”, it also gives developers the ability to develop almost any kind of chatbots with it.

Depending on which of the two pipelines recommended by Rasa Technologies GmbH is used, Rasa NLU supports either all languages or English, German, Portuguese, Italian, Spanish, Dutch, and French (Rasa Technologies GmbH, 2018k).

## Rasa Core

Rasa Core is an open source chatbot framework equipped with machine learning-based dialogue management that predicts the next best action based on NLU input, training data and conversation history. The Rasa Core thus decides what happens next in this conversation. Rasa Core uses a special kind of Recurrent Neural Network that is implemented in the Python

library Keras to predict the next action. (Gaurav, 2018; Rasa Technologies GmbH, 2018n; Williams & Zweig, 2016) Figure 12 illustrates the dialog handling using Rasa Core.

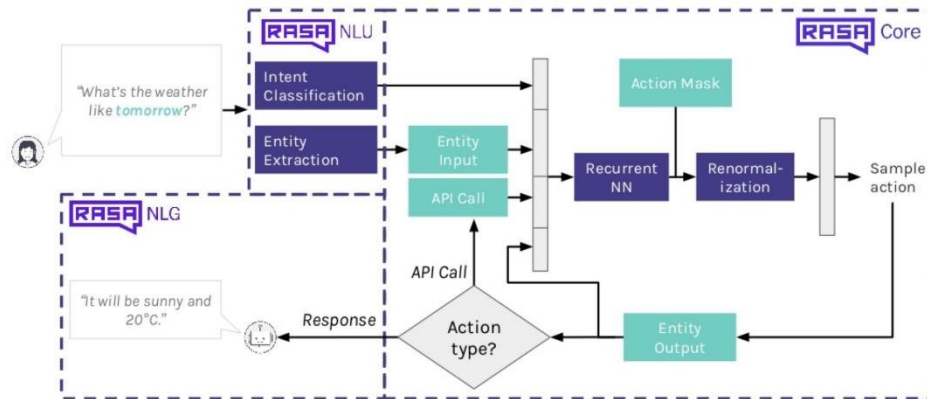


Figure 12 . Dialogue handling using Rasa Core (Petraityte, 2018b, p. 20)

The operational loop shown in Figure 12 starts when the user asks his question. This input is then subjected to an intent classification and an entity extraction. Subsequently, feature vectors are formed in the grey bars, which accept inputs from four sources. These are then passed on to a Recurrent Neural Network, which has the ability to remember past observations for as long as it likes. In the “Actions Mask”, the code of the developer is output, which masks actions due to a not given state. In the renormalization, the mask is used by clamping masked actions to a zero probability. In the step “Sample Action” an action is selected from the probability distribution. The identity of the selected template action is then passed to the developer code “Entity Output”, which is replaced in all template entities. Then it is checked if it is an API text. If so the API call is called in the developer code, otherwise, the action is shown to the user and the cycle starts all over again. (Williams & Zweig, 2016, p. n.d.)

## Rasa Platform

Besides the free open source toolset of Rasa Technologies GmbH, there is also a paid Rasa product called Rasa Platform. As shown in Figure 13, the Rasa Platform complements the Rasa Stack with additional APIs and a user interface to manage them.



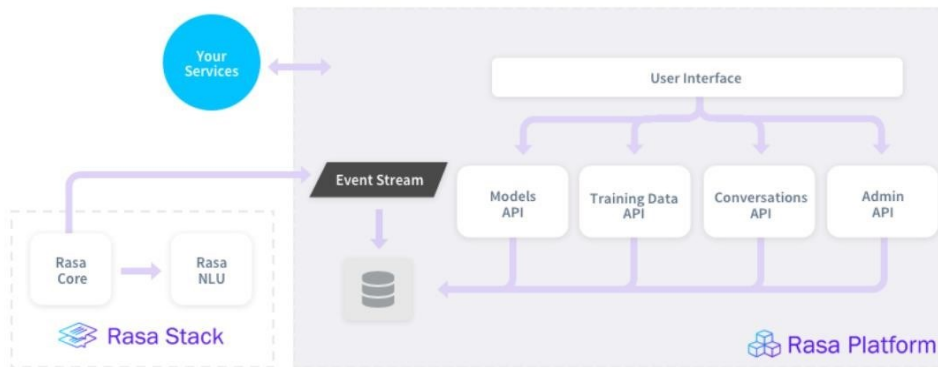


Figura 13 Rasa Platform diagram (Rasa Technologies GmbH, 2019b)

In addition, Rasa Platform customers enjoy more efficient and faster customer support from Rasa Technologies GmbH via the so-called Customer Success Program. It also aims to simplify the training of the machine learning models across business functions, thereby enabling faster iteration cycles (Rasa Technologies GmbH, 2019a, 2019b).

The Rasa platform will not be used in this thesis, it is nevertheless mentioned here for documentary reasons.

## CHATBOT DEVELOPMENT AND IMPLEMENTATION

This chapter explains the key findings, system behavior, and actions that were observed during the development and implementation of a chatbot. The examples, procedures, and scenarios mentioned in this chapter are mostly based on the observations, self-experiments and experiences of the author, the concrete suggestions from expert circles taken from tutorials or forums, as well as from the manufacturer documentation and manufacturer software repositories. Occasionally, references are also made to theoretical approaches and fundamentals.

The practical tests and research described in this chapter were all performed in the same system environment. All test results, operating, system and software behavior mentioned in these subchapters are only valid in an identical system environment. The hardware, operating system

and software specifications of the system environment used can be found in appendix 3.

The use of Python commands in the Linux shell for the execution of Rasa library modules was omitted. Instead, the author decided to create and use Python scripts, which bundle the necessary steps and execute them with a single command. The executed Linux commands are mentioned in the following subchapters in a grey box. The scripts used can be found in the appendices 4, 5, 6 and 7.

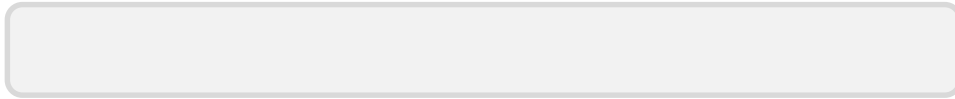
## Software installation

Before installing the Rasa Stack a Python environment is required which is why Anaconda has been downloaded and installed. After installing Anaconda, the Python version was downgraded to 3.6.4, because the later used Python package TensorFlow is not compatible with Python 3.7. It currently only supports Python versions 2.7, 3.4, 3.5 and 3.6 (TensorFlow Community, n.d.). The downgrade can be done using the following command:

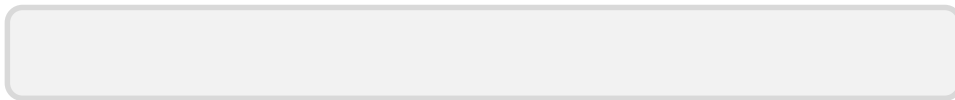
```
rasa:~$ conda install python=3.6.4
```

After the downgrade of Python the three Ubuntu packages `python-dev`, `git-core` and `build-essential` were installed and the system was updated. If the installations went well, the next step is the installation of the required Python packages. Among them are also the Rasa Stack packages `rasa_nlu` and `rasa_core`. These Python packages can either be installed individually by command or all together by reading them from a file and installing them one after the other. The list of used packages and their versions can be found in appendix 5. When installing them, it is important to make sure that all packages are installed correctly, because many of the packages are interdependent. Changes to package versions can automatically result in changes to other package versions, resulting in incompatibility because versions are not supported. The required Python packages can be installed with the following command if they are listed in the `requirements.txt` file:

After that, the language model has to be installed. “The language model is going to be used to parse incoming text messages and extract necessary information” (Petraityte, 2018a). The following command performs the download and installation of the English language model spaCy:



Finally, the `rasa-nlu-trainer` can be installed. This is a user interface that makes generating training data much easier. Since it is a JavaScript-based application, the packages `npm` and `node.js` must be installed first. Then `rasa-nlu-trainer` can be installed with the following command:



```
rasa:~$ npm i -g rasa-nlu-trainer
```

## NLU model

After the installation of the required software, the creation and training of the NLU model can be started. To teach the chatbot how to understand unstructured human language the training of an NLU model is necessary. The trained Rasa NLU model receives constructed text messages and returns data in form of intents and entities, which the chatbot will understand. Since intent classification and entity extraction tasks are machine learning problems, it is necessary to have training data. Figure 14 illustrates the procedure of intent classification and entity extraction. (Petraityte, 2018a)

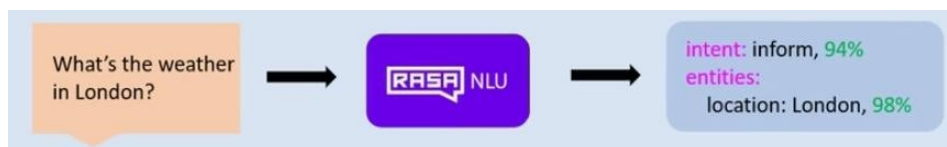


Figura 14 Intent classification and entity extraction (Petraityte, 2018a)

## Data format

The training data can be saved in two different formats. This can either be in a Markdown file or in a JavaScript Object Notation (JSON) file and it can take place either in a single file or in several files located in one directory. (Rasa Technologies GmbH, 2018q)

The use of multiple files can be useful for a large number and variety of training data. In this way, the training data files can be grouped according to topics, for example, which has advantages in the organization of the data and makes it easier to share data across projects. It also makes it easier for the responsible person to manage the training data if several topics such as school, student accommodation and small talk are separated. If several files containing training data are used, the entire directory in which the files are located is specified during the training. If only one data file with training data is used, no directory but only this one file must be specified. Since the number of data used in this work is still manageable, there is no need to use several training data files. (Rasa Technologies GmbH, 2018q)

The choice of format does not matter that much, because it is only used in the beginning when creating and training the NLU model. As soon as the interactive training is carried out, the training data will be saved as a Markdown anyway.

Markdown is the simpler format because it is more readable for people but also easier to write. The data format chosen in this thesis is JSON because using the Rasa NLU Trainer tool, unlike Markdown, allows the use of a graphical user interface which should be shown in this thesis. This graphical user interface can be used to enter new training data or review and modify existing data. Figure 15 illustrates the graphical user interface in the browser with some example training data. (Rasa Technologies GmbH, 2018q)

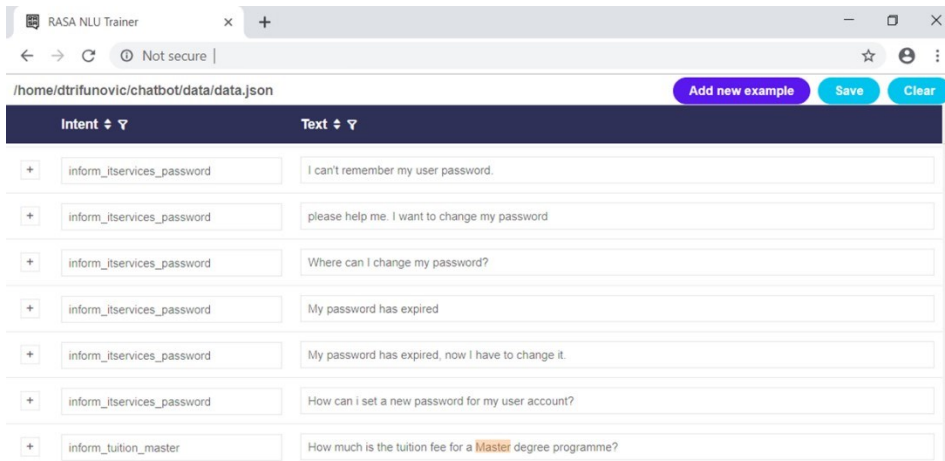


Figura 15 Entering training data using the Rasa NLU Trainer

The condition for using the Rasa NLU Trainer is the use of a JSON-based training data file in a native Rasa NLU format and the opening of an arbitrary port, which is then accessed via the Transmission Control Protocol (TCP). The following Linux command starts the Rasa NLU Trainer listening on port 5005:

```
rasa:~$ rasa-nlu-trainer -p 5005 -s data.json
```

If the parameters `-p` and `-s` are omitted, any port is selected and the first JSON file found by the system as a valid data file is used. Finally, if multiple training data files are used, omitting the `-s` parameter can lead to confusion.

## Data entering

The data entering can either be done in the Linux terminal itself or the installed Rasa NLU trainer can be used. Before the Rasa NLU Trainer can be started, a JSON file must be created and an example made that resembles the one in Figure 16.

```
(base) dtrifunovic@rasa:~/chatbot$ more data/data.json
{
  "rasa_nlu_data": {
    "common_examples": [
      {
        "text": "Hello",
        "intent": "greet",
        "entities": []
      },
      {
        "text": "hello",
        "intent": "greet",
        "entities": []
      }
    ]
  }
}
(base) dtrifunovic@rasa:~/chatbot$ █
```

Figura 16 . Entering training data in Linux terminal

Once the file has been created, Rasa NLU Trainer can be used to create additional examples. Entities can also be entered via the graphical user interface. In order to ensure a clean and error-free training later on, it requires a certain number of examples per intent. Although two examples per intent are sufficient to train the model, they are still not enough to make good statements regarding the intent classification. That is why for each of the ten intents used, ten training examples were captured. With the extension to 35 intents, at least 16 examples were created per intent, so that no warning messages were displayed and that the NLU model could be trained correctly.

## Synonyms

In addition to the training data, synonyms can also be added. This allows the chatbot to recognize the content despite missing keywords by using synonyms. The synonyms can be entered like the training data in the data file. Figure 17 shows some examples of the synonyms used for this thesis. (Rasa Technologies GmbH, 2018q)

```

"entity_synonyms": [
  {
    "value": "Accommodation",
    "synonyms": [
      "Student apartment",
      "student apartment",
      "apartment",
      "housing",
      "student housing",
      "living place"
    ]
  },
  {
    "value": "Degree Programme",
    "synonyms": [
      "degree programme",
      "degree course",
      "study course",

```

Figura 17 Using synonyms for training data

If no synonyms are used and nothing is specified in the data file, the warning message “UserWarning: Failed to load synonym file from ...” may appear when training or validate the NLU model. According to DrissnerSchmid (2018a), it is only a warning, which has no influence on the functionality. The message can thus be ignored, corrected by an empty specification in the data file or by the creation of an empty JSON file. The latter can be done by executing the following command:

```

rasa:~$ echo "{}" > models/nlu/default/hamknlv/entity_synonyms.json

```

To use the defined synonyms, it is necessary that the pipeline contains the `ner_synonyms` component (Rasa Technologies GmbH, 2018q).

## NLU model configuration

Before the training of the NLU model can be started a configuration file is needed. In this configuration file, configuration parameters are given that define how the features are extracted from the text input and how the NLU model is trained. Figure 18 shows the contents of the configuration file `config_spacy.json`. (Rasa Technologies GmbH, 2018o)

```
(base) dtrifunovic@rasa:~/chatbot$ more config_spacy.json
{
  "language": "en",
  "pipeline": "spacy_sklearn",
  "path": "./models/nlu",
  "data": "./data/data.json"
}
(base) dtrifunovic@rasa:~/chatbot$
```

Figura 18 Defining the NLU model configuration

The content shows the used pipeline `spacy_sklearn`, the language, the directory where the NLU model will be stored after the training and the directory where the training data is located.

## Pipeline

The two main pipelines are `spacy_sklearn` and `tensorflow_embedding`. The main difference between them is that pre-trained word vectors of fastText or GloVe are used by the `spacy_sklearn` pipeline. But not by the pipeline `tensorflow_embedding`. The TensorFlow pipeline adapts them specifically for the user's data set. Rasa Technologies GmbH recommends to use the pipeline `spacy_sklearn` if less than 1,000 training examples are used. If, on the other hand, 1,000 or more labeled utterances are used, the pipeline `tensorflow_embedding` is recommended. There are also the pipelines `mitie_sklearn` and `mitie`. They use MITIE as a word vectors' source. Rasa technologies GmbH do not recommend to use these because they could be deprecated in a future release. (Rasa Technologies GmbH, 2018c)

As already mentioned, it depends on the pipeline chosen which languages the Rasa NLU supports. With `tensorflow_embedding` every language is supported because user-defined word embeddings are trained. `Spacy_sklearn`, on the other hand, supports the already mentioned ones. However, Rasa Technologies GmbH offers the possibility to add languages on its own when `spacy_sklearn` is used. (Rasa Technologies GmbH, 2018k)

## NLU model training

For the training of the NLU model the Python script `nlu_model.py`, which is in appendix 4, was used. The script contains the two functions `train_nlu()` and `run_nlu()`. After executing the function `train_nlu()` the NLU model is



created in the corresponding directory and trained. Figure 19 shows the console output after the successful training.

```
(base) dtrifunovic@rasa:~/chatbot$ python nlu_model.py
Fitting 2 folds for each of 6 candidates, totalling 12 fits
[Parallel(n_jobs=1)]: Done 12 out of 12 | elapsed: 4.5s finished
(base) dtrifunovic@rasa:~/chatbot$
```

Figura 19 Successful training of the NLU model

To be sure, the directory can also be checked to see if it contains the files it needs. Figure 20 shows a list of the files in the directory of the created NLU model.

```
(base) dtrifunovic@rasa:~/chatbot$ ls -l models/nlu/default/hamknl/
total 2044
-rw-rw-r-- 1 dtrifunovic dtrifunovic 52503 Mar 1 21:35 crf_model.pkl
-rw-rw-r-- 1 dtrifunovic dtrifunovic 904 Mar 1 21:35 entity_synonyms.json
-rw-rw-r-- 1 dtrifunovic dtrifunovic 1921320 Mar 1 21:35 intent_classifier_sklearn.pkl
-rw-rw-r-- 1 dtrifunovic dtrifunovic 2549 Mar 1 21:35 metadata.json
-rw-rw-r-- 1 dtrifunovic dtrifunovic 2 Mar 1 21:35 regex_featurizer.json
-rw-rw-r-- 1 dtrifunovic dtrifunovic 101602 Mar 1 21:35 training_data.json
(base) dtrifunovic@rasa:~/chatbot$
```

Figura 20 List of needed Rasa NLU model files

## NLU model testing

After the NLU model has been created and trained, it can be tested. To test the training performed, it is necessary to load the model. For this, the function `run_nlu()` from the script `nlu_model.py` can be used. This function performs an intent classification and entity extraction to a question predefined in the script that differs from the training data. Subsequently, the results will be given. The first thing that is given is what the model thinks, what the intent of the predefined input message is. Then, a confidence level is shown, which states how good the intent classification of the model was. This is followed by entity extraction if an entity has been found, and the confidence levels of the remaining intents. So the results of the intent classification are also provided for all other intents that are in the training data. These are listed in a ranking. Figure 21 shows the results after testing the NLU model. The question defined in the script can also be seen at the end of the given results.

```
(base) dtrifunovic@rasa:~/chatbot$ python nlu_model.py
{'intent': {'name': 'inform_responsibility_BIT', 'confidence': 0.3283491359752167}, 'entiti
ties': [{'start': 35, 'end': 46, 'value': 'business it', 'entity': 'DegreeProgramme', 'co
nfidence': 0.979056052899517, 'extractor': 'ner_crf'}], 'intent_ranking': [{'name': 'info
rm_responsibility_BIT', 'confidence': 0.3283491359752167}, {'name': 'inform_contact_hops'
, 'confidence': 0.11196223454273682}, {'name': 'inform_responsibility_BA', 'confidence':
0.04113952015161588}, {'name': 'inform_responsibility_nursing', 'confidence': 0.034081295
898732364}, {'name': 'inform_scanner', 'confidence': 0.028447567872422107}, {'name': 'inf
orm_responsibility_design', 'confidence': 0.02831397728012352}, {'name': 'inform_accommod
ation_equipment', 'confidence': 0.02818637336700294}, {'name': 'inform_printer_scanner_pa
yment', 'confidence': 0.025077289399669973}, {'name': 'inform_problem_hops', 'confidence'
: 0.024151016109228575}, {'name': 'inform_tuition', 'confidence': 0.02261666175367568}],
'text': 'Tell me who is responsible for the Business IT Course?'}
(base) dtrifunovic@rasa:~/chatbot$
```

Figura 21 Successful testing of the NLU model

Figure 21 shows that the correct intent was recognized and a confidence level of 0.3283. However, the confidence level for a question taken from the training data is 0.7239. Thus the training data could still be extended by some utterances to improve the model. These results can later be used when creating a dialogue management model.

## Dialogue management model

Once the Rasa NLU model has been created and trained, the Rasa Core dialogue management model can be created and trained. The dialogue management model is the second part of the chatbot's architecture. It is used to predict what response or action the chatbot should make based on the context and the conversation's state. Those actions could be basic text responses, database queries or API calls. Figure 22 illustrates the procedure for predicting the next action using the Rasa Core dialog management model. (Petraityte, 2018a)

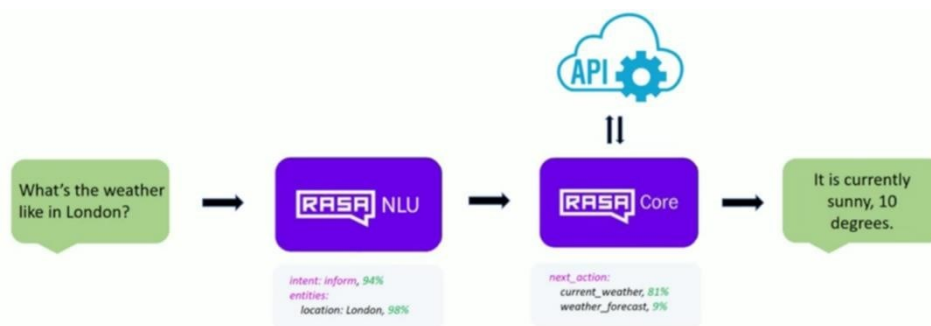


Figura 22 Rasa Core predicts next actions and steps (Petraityte, 2018a)

Two files are required to create the dialogue management model. One contains the definitions of the domain and the other the stories.

### Domain format

According to Rasa (2018f), the domain defines the universe in which the bot works. It indicates the entities, intents, slots, and actions the chatbot should know. As an option, it can also contain answer templates the bot can use. While the intents and entities have already been defined in the training data and are only mentioned in the domain file, the actions and templates are newly recorded. Slots are the memory of the bot. They function as key value memories in which both information about the outside world and information about the user, such as his home town, can be stored. Slots are intended to influence the progress of the dialog. There are different slot types for different behaviors. Examples are float, bool, categorical, text, list or unfeaturized. (Rasa Technologies GmbH, 2018f)

It was decided that the answers of the chatbot would be hardcoded in the templates section instead of being retrieved from appropriate systems via APIs.

Nevertheless, there is the possibility to get information and whole answers to questions from the surrounding systems instead of coding them hard. The CMS WordPress used allows developers to read, create new and edit existing website content via Representational State Transfer (REST) API. With the help of custom actions written in Python, the necessary information can be retrieved from the CMS. For content that needs to be changed periodically, such as the deadline for study enrolments, this option might be better than hard coding. (Rasa Technologies GmbH, 2018a; WordPress Foundation, n.d.)

The next step is to define a domain. For that a domain file containing the intents, entities, actions, templates, and slots is created. Figure 23 shows a part of the content of the domain file used.

```

slots:
  Location:
    type: text
  DegreeProgramme:
    type: text
  DegreeLevel:
    type: text

intents:
- greet
- goodbye
- mood
- inform_accommodation_address
- inform_accommodation_price
- inform_scanner
- inform_printer

entities:
- Location
- DegreeProgramme
- DegreeLevel

templates:
  utter_greet:
    - 'Hey there, how can I help you?'
    - 'Hello, how can I help?'
  utter_goodbye:
    - 'Bye bye'
    - 'See you later.'
  utter_mood:

```

Figura 23 Examples of definitions within the domain

The actions that are not visible in figure 23 may contain defined utterances or self-programmed actions that forward e-mails or perform API calls.

## Tone of voice

The answers of a chatbot can be short and concise or even more detailed. In both cases, the tone of voice can be important to the user. The pie chart in Figure 24 shows the result of the question about the most important characteristic of a chatbot, which was last asked in the survey conducted.

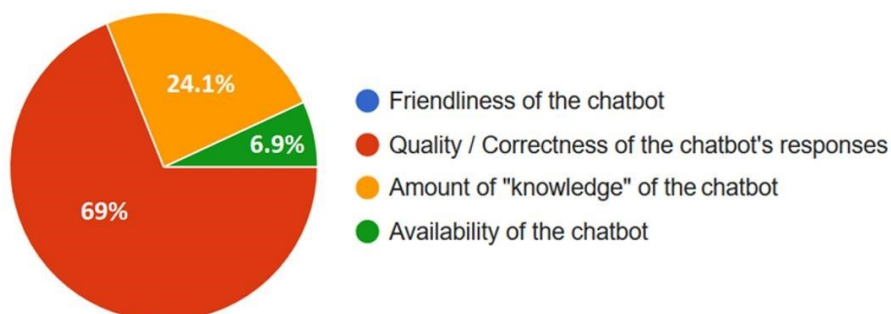


Figura 24 Most important characteristic of a chatbot

The student information obtained from the survey showed in figure 24 that almost 70% felt that the most important characteristic of a chatbot should be the correctness of its answers. About a quarter felt that the amount of knowledge of the chatbot had to come first. The remaining 6.9% thought that the availability of the chatbot was the most important one. However, none of the respondents found friendliness to be the most important feature. Nevertheless, this does not mean that the answers given by the chatbot do not have to be friendly.

Statista surveyed 5,000 people on the preference of the personality type of a chatbot and presented the results as a bar chart in Figure 25 (Statista GmbH, 2017a).

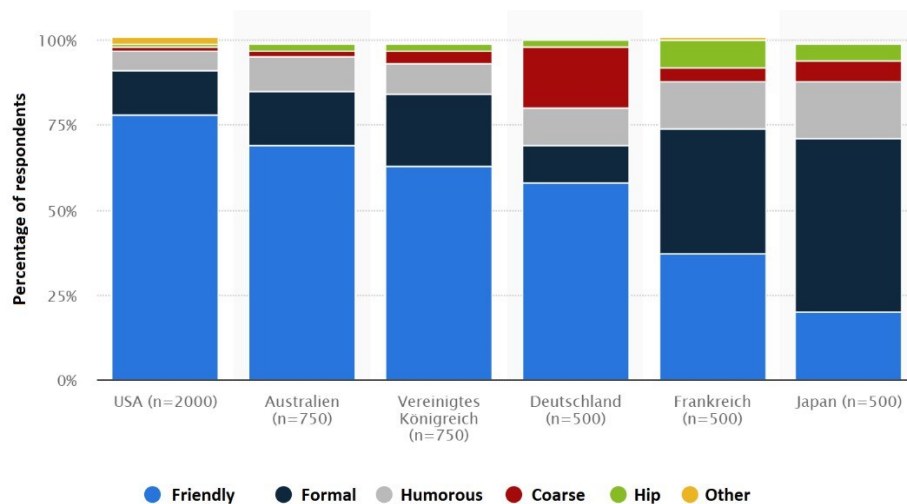


Figura 25 Preferred personality of a chatbot by country (Statista GmbH, 2017a)

The navy blue and azure blue areas shown in Figure 25 show that friendly and formal chatbot personalities are preferred.

Although the personality of a chatbot is not perceived as the most important criteria, the answers of the chatbot should nevertheless have a friendly formal character. First, because the results of the interviewed students do not make a statement about the importance of the characteristic in itself, but compare the importance of the characteristics with each other. However, this does not mean that one or more of them are unimportant. On the other hand, Statista's statistics (2017a), which include the results of 5,000 people, are representative and effectively built on the personality of the chatbots.

When choosing the answers, attention was paid to both the correctness and the personality of the chatbot. It was tried to write them in a formal and friendly way.

### Story data format

A Rasa Core dialogue system's training example is called story. These training examples are written in a Markdown file. The stories are assigned to the intents used. This ensures that the chatbot knows later with which action it has to react at which intent. Multiple actions can be assigned to a single intent. A story is, therefore, a conversation between a user and a chatbot. The following figure 26 shows a part of the stories used in the corresponding format. (Petraityte, 2018a; Rasa Technologies GmbH, 2018p)

```
## Stateless stories
## Story 01
* greet
  - utter_greet

## story 02
* goodbye
  - utter_goodbye

## story 03
* mood
  - utter_mood

## story 04
* inform_printer_scanner_payment
  - utter_price_printer
```

*Figura 26 Examples of stories within the story file*

In order to get good stories, Petraityte (2018a) suggests to ask acquaintances or future users of the chatbot what kind of conversations they would have with the chatbot. Thus, questions could be found that did not occur to the creator, which can be used as training examples. Petraityte (2018a) also recommends the use of a Rasa Core features called online training. In addition to helping to create story data, it also trains a real-time dialogue management model. (Petraityte, 2018a)

## Online training

Appendix 5 contains the Python script `train_init.py`, which creates and trains the Rasa Core dialogue management model based on the previously created training data. At this point, it should be noted that various parameters are used in the script for the training, which should be adjusted according to the size of the training data. Figure 27 shows the successful training of the dialogue management model.

```
Epoch 96/100
3213/3213 [=====] - 0s 109us/step - loss: 0.2680 - acc: 0.8820
Epoch 97/100
3213/3213 [=====] - 0s 111us/step - loss: 0.2772 - acc: 0.8883
Epoch 98/100
3213/3213 [=====] - 0s 111us/step - loss: 0.2662 - acc: 0.8802
Epoch 99/100
3213/3213 [=====] - 0s 113us/step - loss: 0.2706 - acc: 0.8852
Epoch 100/100
3213/3213 [=====] - 0s 109us/step - loss: 0.2815 - acc: 0.8824
INFO:rasa_core.policies.keras_policy:Done fitting keras policy model
INFO:rasa_core.agent:Persisted model to '/home/dtrifunovic/chatbot/models/dialogue'
(base) dtrifunovic@rasa:~/chatbot$
```

*Figura 27 Successful training of the dialogue management model*

Depending on the parameters used and the amount of training data, the training process may take longer or shorter. After the training has been completed successfully, it can be checked in the corresponding directory whether the files of the dialogue management model are located there.

At this point, online training can be launched. This requires another Python script called `train_online.py`, which can be found in appendix 6. The online training is an interactive learning mode, where the developer provides feedback while talking to the chatbot. In this way, it is possible to examine what the bot is already capable of and subsequently correct any mistakes that the bot makes. If the bot does not know how to do something, it can be taught within the online training. (Rasa Technologies GmbH, 2018j). At this point, it should be noted that in the script for online training various parameters are used, which should be adjusted depending on the size of the training data. Figure 28 shows the successful launch of the online training.







Figure 29 shows that the user's message was correctly classified during online training. Then the bot suggested to reply with `utter_greet`. In the end, the bot wanted to continue with the `action_listen`. Since all three suggestions were correct, they were left as they were and no correction had to be applied.

If the online training should be finished, it can be done after each questions passage by pressing the Enter key. Then there are five possibilities what can be done next. Figure 30 shows the five possibilities.

```
? Next user input (Ctrl-c to abort):
? Do you want to stop? (Use arrow keys)
  Continue
  Undo Last
  Fork
   Start Fresh
  Export & Quit
```

Figura 30 Selecting action after training

If one decides for the export and the termination, then one is asked afterwards to indicate the files in which the training data has to be stored. Figure 31 illustrates this.

```
Current slots:
  DegreeLevel: None, DegreeProgramme: None, Location: None
-----
? The bot wants to run 'action_listen', correct? Yes
? Next user input (Ctrl-c to abort):
? Do you want to stop? Export & Quit
? Export stories to (if file exists, this will append the stories) data/stories.md
? Export NLU data to (if file exists, this will merge learned data with previous training examples) data/data.json
? Export domain file to (if file exists, this will be overwritten) domain.yml
```

Figura 31 Provide the files in which the training should be saved

During the online training, Rasa Core records the current conversation and some similar conversations from the training data. This should help the user to keep track of where he is at any given moment. The visualization can be viewed in the browser at <http://localhost:5005/visualization.html> as soon as the online training was started. Figure 32 shows an example of the visualization during interactive learning. (Rasa Technologies GmbH, 2018j)

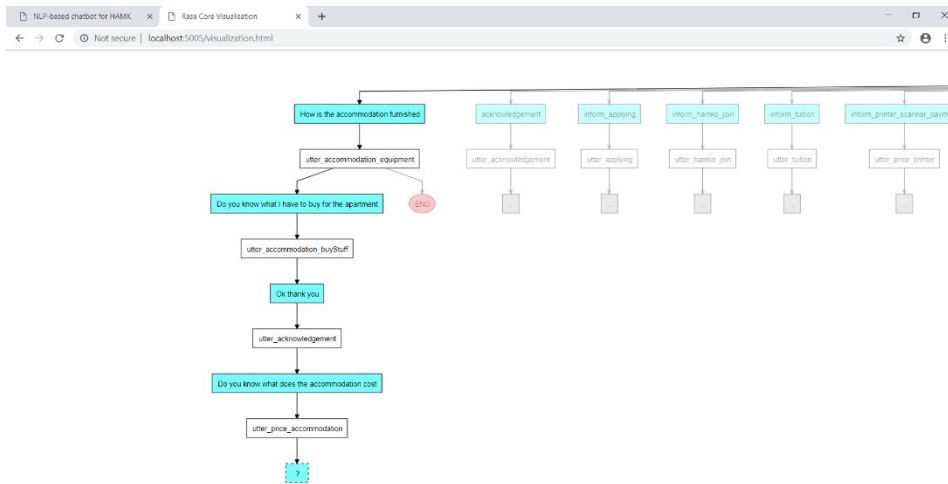


Figura 32 Visualization of the conversation during interactive learning

In addition to the visualization of interactive learning, stories can also be visualized. This can be helpful if an overview of the conversation paths described in a story file is needed. But it can also be used to simplify debugging and discussions about the bot flow. (Rasa Technologies GmbH, 2018e) With the following Linux command stories can be visualized:

```
rasa:~$ python -m rasa_core.visualize -d hamk_domain.yml -s
data/stories.md -o generatedGraph.html
```

After the execution of the command above the file, generatedGraph.html is available, which can be opened in the browser afterwards. Figure 33 shows a visualization of the conversation paths of a story file, which was created with the above command. It indicates the end result of the training completed so far. The content in the case of figure 33 is irrelevant. The figure is intended to illustrate how complex the paths can become after just a few training sessions.

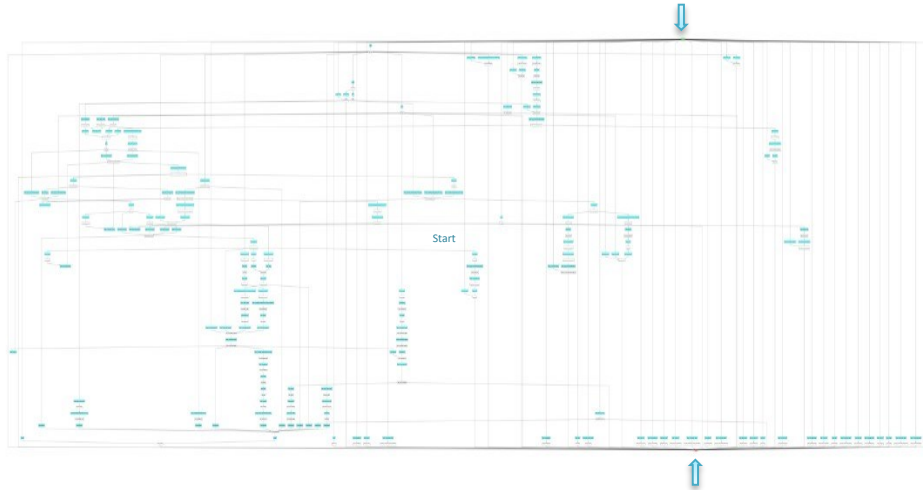


Figura 33 Visualization of the conversational paths of a story file

End

## Creating chatbot

At this point, everything is done that is necessary to create a chatbot that can communicate with someone. After this several possibilities can be considered. In this thesis, only two are used. On the one hand a command line based chatbot, with which it is possible to chat within the Linux console and which at the same time ensures that the just created project works. And secondly, a chatbot embedded on a website, which can be used by other users.

### Chatbot on the Linux command line

Appendix 7 contains the Python script `dialogue_mgmt_model.py`, which starts a chatbot inside the Linux terminal. After the execution of the script, questions can be asked to the chatbot. Figure 34 shows such an example.

```

INFO:root:Rasa Core server is up and running on http://localhost:5005
Bot loaded. Type a message and press enter (use '/stop' to exit):
Hi Lisa
Hey there, how can I help you?
127.0.0.1 - - [2019-03-03 21:51:33] "POST /webhooks/rest/webhook?stream=true&token=
230 0.106548
I need your help
Tell me what is on your mind.
127.0.0.1 - - [2019-03-03 21:51:37] "POST /webhooks/rest/webhook?stream=true&token=
229 0.015930
how expensive is a sports pass?
The prices are as follows: The members of HAMKO pay 45 Euro per semester or 80 Euro
year. Other students of HAMK pay 55 Euro per semester or 100 Euro per academic year.
udents pay 65 Euro per semester or 120 Euro per academic year.
127.0.0.1 - - [2019-03-03 21:51:47] "POST /webhooks/rest/webhook?stream=true&token=
457 0.017297
Thank you
You are welcome :)
127.0.0.1 - - [2019-03-03 21:51:52] "POST /webhooks/rest/webhook?stream=true&token=
218 0.016120

```

*Figura 34 Chat with a chatbot inside the Linux terminal*

Figure 34 shows that the chatbot works and is therefore ready for implementation on a website.

### Chatbot on the own website

To operate the developed chatbot on an own website it is necessary to have an own website. For this purpose, a ready-made website template was downloaded from the internet. After the website template was adapted, an Apache web server was installed. Before the web server was started, the created website was copied to the corresponding server directory, and the permissions of the files were adapted. With a working website now available on a web server, the implementation of the chatbot can begin.

Rasa offers a wide range of ways to connect the created chatbot with the outside world. Among the third-party channels offered by Rasa are various communication, voice and chat platforms from renowned manufacturers such as Facebook, Microsoft, Slack, Telegram or Cisco. If one wants to have a chatbot widget on one's own website, the GitHub projects Chatroom and Rasa Webchat referenced on the Rasa website can be used. Rasa Webchat uses sockets while the Chatroom uses regular HTTP requests. Both products can be used via the RestInput or the CallbackInput channel. (Rasa Technologies GmbH, 2018b)

For this thesis, the product Chatroom of the company Scalableminds GmbH (2019) was chosen, without carrying out an evaluation. First, the JavaScript chatroom.js was embedded into the Hypertext Markup Language (HTML) of the website and configured to be visible on the web page (Scalableminds GmbH, 2019). Figure 35 shows the embedding at the bottom of the website.

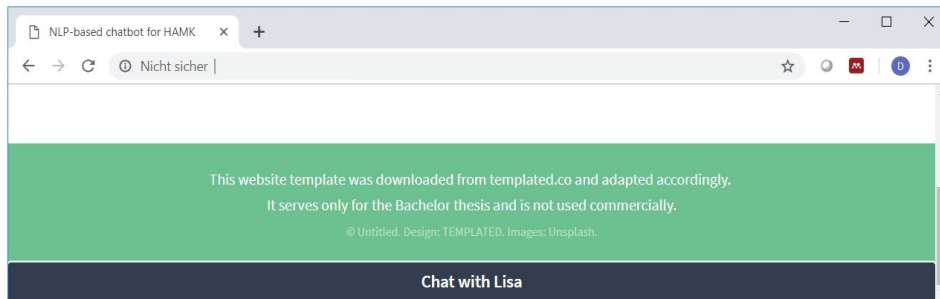


Figura 35 Embedding the chatbot widget on the website

The JavaScript file and the Cascading Style Sheets (CSS) file are retrieved from the web and integrated into the HTML. However, it is also possible to download these two files, store them on the server and adapt them to the styles, guidelines, and specifications given by The AI Institute.

After downloading the GitHub Project Chatroom, copying the Python script `bot_server_channel.py` into the Rasa Core project and creating the needed files `credentials.yml` and `endpoints.yml`, the chatbot service could be started with the following command:

```

rasa:~$ python -m rasa_core.run --core models/dialogue/ --nlu
models/nlu/default/hamknl/ --endpoints endpoints.yml --credentials
credentials.yml

```

Figure 36 shows a working chatbot widget on the website that can handle voice messages and hyperlinks such as website links or the uniform resource identifier scheme for e-mail addresses `mailto`.

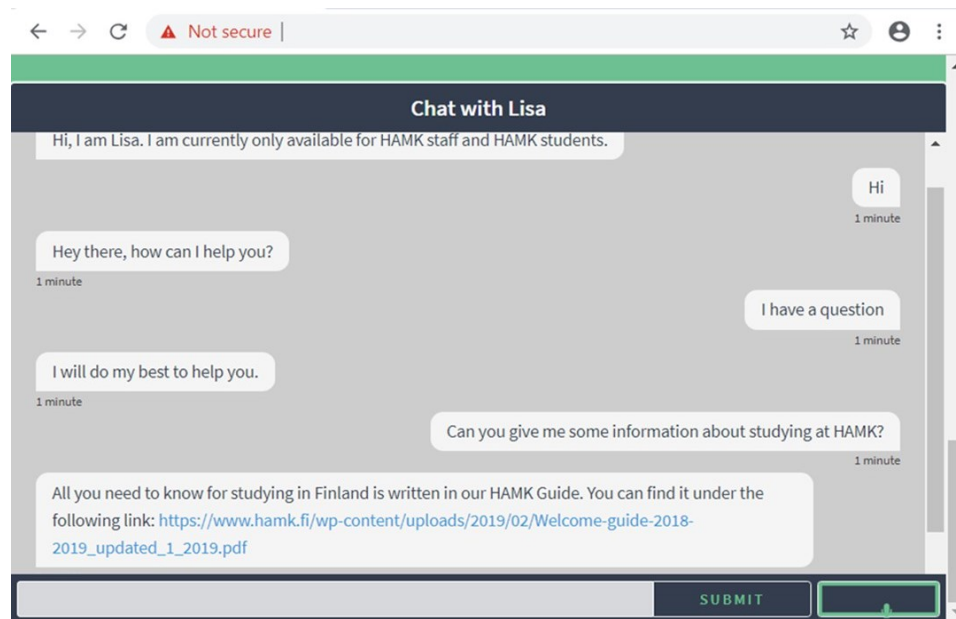


Figura 36 Chat with a chatbot implemented on a website

## Chatbot operation

The following chapter contains suggestions and ideas that came up during the development of the chatbot and could be useful for the further operation of the prototype. While some of them were realized and tested in the developed prototype, others were only discussed theoretically.

## Logging

As mentioned earlier, Rasa Technologies GmbH believes (2018i) that real conversations between AI and people are more important than hypothetical approaches to get good training data. Various methods can be used to retrieve training data. Logging conversations is one of them. By using Chatroom, conversations are automatically logged without having to make any configurations. A JSON file is created, which is continuously updated, even if the chatbot service has been restarted on the server. By using an appropriate formatter, the JSON content can be made readable for people. Figure 37 shows a conversation that has taken place and has been made legible by using an online JSON formatter.





## Fallback action

Rasa Technologies GmbH offers with Rasa Core the possibility that the chatbot can fall back on a fallback action such as “I am sorry, but unfortunately, this is not in my area of knowledge” in case of ignorance. For the chatbot to do this, the FallbackPolicy must be added to the existing policies. “The fallback action will be executed if the intent recognition has a confidence below `nlu_threshold` or if none of the dialogue policies predict an action with confidence higher than `core_threshold`” (Rasa Technologies GmbH, 2018g). The thresholds and fallback action can be modified as parameters in the policy configuration file. Instead of changing this in the policy configuration file, the `action_default_fallback` within the Python script can also be captured, which then returns an `utter_default` template to the user as a response. Figure 38 shows the fallback action using an example. (Rasa Technologies GmbH, 2018g)

```
Bot loaded. Type a message and press enter (use '/stop' to exit):
Hi Lisa
Hey there, how can I help you?
127.0.0.1 - - [2019-03-10 15:30:01] "POST /webhooks/rest/webhook?stream=true&token=
I need NBA tickets for the match next saturday. Can you order them for me?
I am sorry, but my creator did not teach me about that yet :(
127.0.0.1 - - [2019-03-10 15:30:31] "POST /webhooks/rest/webhook?stream=true&token=
ok, thank you anyway
You are welcome :)
```

*Figura 38 Fallback action of the chatbot*

If questions are asked that the chatbot does not know, or if the chatbot has not yet been trained well enough, an incorrect intent could be selected in the intent classification, which leads to an incorrect answer from the chatbots. To avoid this, fallback actions should be used.

## Training process

The training effort of the chatbot is an ongoing, never-ending process. After a first version of the chatbot has been created, it can be trained continuously using the Python scripts already mentioned. As soon as the Rasa NLU training data has been extended, the NLU model can be updated using the script `nlu_train.py`. The same applies to the dialogue management model. As soon as changes have been made to the Rasa Core training data,



the dialogue management Model can be updated using the script `train_init.py`.

### Migration guide

If Rasa Stack version changes are carried out, this can result in adjustments in various areas. On the one hand, it must be ensured that the dependencies of the installed Python packages are guaranteed. On the other hand, the created scripts have to be checked, because it is possible that imported modules have been renamed, used functions got a new name or have to be parameterized differently. The Migration Guide of Rasa Technologies GmbH (2018I) provides a remedy for this. This guide contains information about changes between the main versions and about how a version migration can be carried out.

## DEVELOPMENT FINDINGS AND RESULTS

The points mentioned in this chapter are results and insights that have emerged from the practical part of this thesis, the development of the chatbot. Because only the development process was described in the previous chapter, they are mentioned in this chapter along with recommendations from the author. The recommendations are based on observations and statements already mentioned in the theory of this thesis.

### Data

The data is the foundation of the chatbot. If the quality of the data is poor, this affects the training as well as the response behavior of the chatbot. For this reason, it is important to ensure that attention is paid to this activity.

For data collection, log files, surveys, creativity techniques, interviews, email requests received, and past requests could be used. It is recommended to use different sources for data collection, as it will be difficult for a single person or a small group to bring many different training data together. Despite the data collection methods used in this thesis, the use of the chatbot by others showed that utterances were used that were not collected and included in the training data. Collecting the data should, therefore, be a repetitive process.

It is recommended that when using the chatbot, the areas of use which the chatbot should cover are analyzed.

### Installation and development

Rasa Technologies GmbH is a company founded in 2016, which has established itself in the last years by designing, developing and operating platforms for the creation of conversational software using open source machine learning toolkits (Bloomberg L.P., 2019; Rasa Technologies GmbH, 2019a). The dynamism of the company can also be felt in its products. Last year there were several new updates for the Rasa Stack. What speaks in favor of the company in terms of agility in development can be somewhat detrimental to the inexperienced user by consulting different sources of information. Version updates may include changes to

library names, module names, and function names. It is therefore recommended that when consulting the Rasa NLU or Rasa Core documentation, the version is looked at and used during installation. Due to the dependencies of the software used and the Python packages installed, complications can occur during the installation of the Rasa Stack if the dependencies between them are not guaranteed. It is therefore not recommended to update the used software, Python libraries or the Rasa Stack during the development process, but to do this either at the beginning or afterwards, using the Migration Guide.

The development of the chatbot is well described on the website of Rasa Technologies GmbH. There are also some tutorials and blogs on the Internet that can be considered for the development process. However, there is a possibility that errors and problems may occur during the development process due to various causes. It may also be possible that the error messages and their solutions are not or only minimally documented on the Internet. Therefore a detailed analysis of the problem and fundamental Python knowledge will be necessary. In most cases, problems can be solved using the Rasa Community Forum and the GitHub project. Questions can be asked about problems or ambiguities, but also answers and solutions to problems that have already occurred can be found there. The Rasa development team reacted quickly to new issues and in most cases offered useful solutions within a short time. The only disadvantage that was found was that problems were at some point called solved by the issue creators without them presenting the concrete solution. Thus, in the case of the same issue, other possibilities had to be sought. However, this is not a problem on the part of Rasa Technology GmbH.

Rasa Technologies GmbH recommends the use of Docker for production deployments, which entails the condition that Docker is used as software. For Rasa Stack, pre-built Docker images are provided on Docker Hub that can be downloaded and used. The Rasa platform is also containerized and can run either in the private cloud or on premise. A container orchestration ready for production is included in the delivery's scope. (Rasa Technologies GmbH, 2018i)

What can also be recommended is a correct and logical naming of intents, entities, and utterances. This can be done, for example, using a naming concept. It simplifies the identification of the intents, entities, and utterances during interactive training. If the number of training data used

is manageable, a naming concept does not necessarily have to be applied. But as soon as the number increases, the overview is quickly lost. As a result, confusion and uncertainty can occur during online training. To avoid subsequent adjustments of training data, a uniform naming scheme can be used right from the start.

With regards to the user experience, it is also useful to think about the length of the chatbot's response. Too long text messages may have a negative impact on the user. Jindal (2017), founder of the technology platform Tars, which makes it possible to build conversational bots without programming knowledge, generally follows the Twitter approach and keeps its messages as big as a tweet.

### Operation and periodic training

The operation of a chatbot has shown that by using it, information can be obtained much more quickly than if the information had to be found via the website. However, this is only the case if the question has been asked grammatically correctly to the chatbot and if the chatbot knows the answer to the question and has been trained to do so.

The difficult part in the author's eyes is to train the chatbot. This responsibility, as well as that of the operation, should be left to the IT Services. On the one hand, because problems during operation can also be tackled directly by the relevant department, and on the other hand, because of the interaction with a Linux terminal, it requires appropriate IT expertise in the areas of Linux, Python and web technologies. The exception could be the use of software with a graphical user interface such as Smart Platform Group's (2017) Articulate, which allows training data to be supplemented and training to be carried out via a browser.

Nevertheless, the operation would be the task of IT, while the training could probably be carried out by other departments. Since such products were not investigated and tested in this thesis, no more detailed recommendations can be given.

As already mentioned in theory, the best training data is not the hypothetical data, but the data that can be collected from conversations with people. It was also mentioned that the training of a chatbot never ends. It is once again recommended at this point that if a chatbot is used, it should be ensured that the training process is a recurring process that takes place regularly.

Weak training or an insufficient amount of data leads to weak chatbot service. This was also noticed in the development of the chatbot. After the number of 12 utterances was increased to over 30, this was also noticeable in the intent classification. Intent scores of new utterances, which were around 0.30 before the extension of the training data, subsequently had values 0.70. However, the number of utterances used per intent varies. The more intents are used, the more utterances per intent are needed. In this thesis, 39 intents were used, each with an average of 17 utterances.

In the training data, the utterances can be equipped with punctuation marks such as coma, dot, question mark or ellipsis. Rasa differentiates strongly when using the punctuation marks when there is not much training data available. Tests showed that different confidence scores were achieved when checking the same utterance with different punctuation. Since nowadays chatting takes place partly without using punctuation, it is recommended to record the same utterances two or more times with different punctuation marks.

## Testing

As shown above, the most important chatbot's characteristic for the students was the correctness of the answers. Since the students are the end users, and thus the acceptance of the chatbot is determined by them, it is important to ensure the quality of the answers. To achieve this, testing the chatbot is a very good way. However, as soon as a first version or a prototype of the chatbot is available, this should be done by people who have nothing to do with creating the training data. As already mentioned in theory, a wide variety of sentence constructions can be used for user inputs, which can lead to weak intent classification results if not enough training data or insufficient training has been completed.

This situation was observed when the created bot was presented to the client and two fellow students for testing. While the author as the creator of the chatbot knew the data model very well and knew exactly which questions had to be asked how to get the desired result, this was not the case for the other three persons. A person independent of the chatbot development does not know what the NLU model looks like. Their inputs to the chatbot will therefore not be based on the defined utterances within the training data but will be entered exactly as they fall into the person's

mind. The omission of keywords, typing errors or excessive reformulations finally showed that the confidence score was so low that it was answered by the utter\_default. And that is good feedback which helps to expand the training data and the training of the chatbot.

For this reason, it is proposed to have the chatbot tested by people who have nothing to do with the development. In order to obtain a wider range of data, it could even be proposed here to have the chatbot tested by people with different demographic characteristics such as age, educational level, country of origin, knowledge of English or degree course.

Treml (2018d) also suggested testing the chatbot by a crowd but in a much wider range. After crowd testing or exploration testing, one gets finally many new utterances which can flow into the training data. (Treml, 2018d)

## CONCLUSION AND RECOMMENDATION

The teaching of the chatbot provided by Rasa Technologies GmbH is done manually by the user. The first step is to collect, record and store training data. Based on this data, the NLU and dialog models are created, which can be extended further. The training takes place via command line input or can be handled by Python scripts. In the interactive learning mode, the chatbot can be given feedback on his reactions during the chat in order to improve its chat behavior. The training is a process that should be repeated at regular intervals. Using third-party software, training data and training processes can also be managed through a graphical web user interface. The chatbot's answers are only as good as the training it receives. The quality of the training, in turn, depends on the training data. The bot showed that, with 39 intents, averaging about 17 utterances each, it was able to classify the intents correctly and give answers quickly and correctly when the utterances were included in the training data. Intents that were rich in training data were also able to come through with a typo. However, to improve the chatbot in such a way that user input with varying vocabulary or omissions of punctuation marks can bypass a fallback action filter, more training and a much higher amount of training data are required.

The manual installation of the Rasa Stack on a server can have its hurdles due to software dependencies. To avoid this, pre-built Docker images can

be used to automate the installation. A working Rasa chatbot can thereafter be connected to various messaging and voice platforms such as Slack, Telegram or Facebook using an API. This enables it to interact with users on these platforms. The integration into the own web page can be accomplished with the help of third-party software without a large effort. The chatbot's answers have been hardcoded and mainly based on the information available on The AI Institute website.

In case of operation, it is recommended to collect further training data and to deal deeper with the chatbots' training. It should also be considered whether to use the free or the commercial version, which offers better customer support. In order to ensure the chatbot's quality it is suggested to elaborate a test procedure and to carry out corresponding tests with it.

## AUTHOR'S REFLECTION

The chatbot prototype's development was successful from the author's point of view. It can be trained, questions answered and embedded into existing environments. The questions could be answered within the defined scope. The answers do not provide a recommendation as to whether the product should be used in the end or not.

In order to make a qualitative statement about it, a product analysis and corresponding evaluation of the competing products is required, which was not examined in this thesis. It can also be said that the research questions could be examined with a different approach in more depth and answered in more detail, but this requires a longer investigation and the application of certain methods and metrics. Nevertheless, the author is of the opinion that the thesis provides a good insight into the topic on which The AI Institute can rely in its future choice of chatbot.



## REFERENCES

- Anaconda Inc. (2019a). Anaconda Distribution. Retrieved February 2, 2019, from <https://www.anaconda.com/distribution/>
- Anaconda Inc. (2019b). Using multiple versions of Python with Navigator — Anaconda 2.0 documentation. Retrieved February 2, 2019, from <https://docs.anaconda.com/anaconda/navigator/tutorials/use-multiple-python-versions/>
- Badertscher, K., Geiger, I. K., Gubelmann, J., Pifko, C., & Romano, R. (2014). *Projektmanagement - Zertifizierung nach IPMA(3.0)-Ebenen D und C* (3rd ed.). Zürich: Compendio Bildungsmedien AG.
- Bansal, N. (2018). How can we improve confidence score of intents. Retrieved March 6, 2019, from <https://forum.rasa.com/t/how-can-we-improve-confidence-score-ofintents/1788>
- Becker, J. H. (2018). Kreativitätstechniken. In *Praxishandbuch berufliche Schlüsselkompetenzen* (pp. 89–102). Berlin, Heidelberg: Springer Berlin Heidelberg.  
[https://doi.org/10.1007/978-3-662-54925-4\\_11](https://doi.org/10.1007/978-3-662-54925-4_11)
- Bihade, S. A., Badhiye, P. A., & Shelke, R. (n.d.). Chatbots: The emulating machines. *IJFEAT - International Journal for Engineering Applications and Technology*, 3(9), 63–67. Retrieved from <http://www.ijfeat.org>
- Bird, S., Klein, E., & Loper, E. (2015). Analyzing Sentence Structure. Retrieved February 23, 2019, from <https://www.nltk.org/book/ch08.html>
- Bloomberg L.P. (2019). Company Overview of Rasa Technologies GmbH. Retrieved from <https://www.bloomberg.com/research/stocks/private/snapshot.asp?privcapId=542520437>
- Botium GmbH. (2019). Botium - new generation testing. Retrieved March 8, 2019, from <https://www.botium.at/index.html>
- Braun, A. (2003). *Chatbots in der Kundenkommunikation*. Berlin, Heidelberg: Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-642-19021-6>
- Braun, D., Hernandez-Mendez, A., Matthes, F., & Langen, M. (2017). Evaluating Natural Language Understanding Services for Conversational Question Answering Systems. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue* (pp. 174–185). Association for Computational Linguistics. <https://doi.org/10.18653/v1/W17-5522>

Britz, D. (2016). Deep Learning for Chatbots, Part 1 – Introduction. Retrieved March 12, 2019, from <https://www.kdnuggets.com/2016/04/deep-learning-chatbots-part1.html>

Castrounis Alex. (2016). Artificial Intelligence, Deep Learning, and Neural Networks, Explained. Retrieved February 17, 2019, from <https://www.kdnuggets.com/2016/10/artificial-intelligence-deep-learning-neuralnetworks-explained.html>

Chakrabarti, C., & Luger, G. (2013). A framework for simulating and evaluating artificial chatter bot conversations. *FLAIRS 2013 - Proceedings of the 26th International Florida Artificial Intelligence Research Society Conference*, 34–39.

Chatbots.org. (n.d.). Chatbot Parry. Retrieved February 24, 2019, from <https://www.chatbots.org/chatbot/parry/>

Colby, K. M. (1975). *Artificial paranoia; a computer simulation of paranoid processes*. Pergamon Press. Retrieved from <https://dl.acm.org/citation.cfm?id=542898>

Davis, P. (2016). *Using Python in an Introductory ODE Course*. Retrieved from [https://patrickdavis.info/assets/Davis\\_PythonODE.pdf](https://patrickdavis.info/assets/Davis_PythonODE.pdf)

Deloitte Touche Tohmatsu Limited. (2016). *Artificial Intelligence Innovation Report*. London. Retrieved from <https://www2.deloitte.com/content/dam/Deloitte/at/Documents/human-capital/artificial-intelligence-innovation-report.pdf>

Döbel, I., Leis, M., Vogelsang, M. M., Neustroev, D., Petzka, H., Riemer, A., ... Welz, J. (2018). *Maschinelles Lernen: Eine Analyse zu Kompetenzen, Forschung und Anwendung*. München: Fraunhofer-Gesellschaft. Retrieved from [https://www.bigdata.fraunhofer.de/content/dam/bigdata/de/documents/Publicationen/Fraunhofer\\_Studie\\_ML\\_201809.pdf](https://www.bigdata.fraunhofer.de/content/dam/bigdata/de/documents/Publicationen/Fraunhofer_Studie_ML_201809.pdf)

Drissner-Schmid, A. (2018a). Failed to load synonyms file entity\_synonyms.json. Retrieved from [https://github.com/RasaHQ/rasa\\_nlu/issues/1146](https://github.com/RasaHQ/rasa_nlu/issues/1146)

Drissner-Schmid, A. (2018b). Handle negation and misspelling.

Elhadad, M. (2012). Parsing - NLP 2012. Retrieved February 23, 2019, from <https://www.cs.bgu.ac.il/~elhadad/nlp12/nlp03.html>

- EliteDataScience.com. (2019). Overfitting in Machine Learning. Retrieved March 9, 2019, from <https://elitedatascience.com/overfitting-in-machine-learning>
- Epping, T. (2011). *Kanban für die Softwareentwicklung* (1st ed.). Berlin, Heidelberg: Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-642-22595-6>
- Feyhl, A. W. (2004). Methoden und Vorgehensmodelle. In *Management und Controlling von Softwareprojekten* (pp. 13–63). Wiesbaden: Gabler Verlag.  
[https://doi.org/10.1007/978-3-322-86755-1\\_2](https://doi.org/10.1007/978-3-322-86755-1_2)
- Finnish National Agency for Education. (n.d.). Studyinfo.fi. Retrieved March 10, 2019, from <https://studyinfo.fi/>
- Gaurav, G. (2018). Understanding and Leveraging Machine Learning in Conversational AI (using Rasa). Retrieved March 10, 2019, from <https://medium.com/datadriveninvestor/understanding-and-leveraging-machinelearning-in-conversational-ai-using-rasa-73d3136e4d16>
- Gauthier, J. (2016). Hybrid tree-sequence neural networks with SPINN. Retrieved February 23, 2019, from <https://nlp.stanford.edu/blog/hybrid-tree-sequenceneural-networks-with-spinn/>
- Geeb, F. (2007). Chatbots in der praktischen Fachlexikographie und Terminologie. *LDVForum: Zeitschrift Für Computerlinguistik Und Sprachtechnologie*, 22(1), 51–70.  
Retrieved from <https://pdfs.semanticscholar.org/0a30/60961ae1dd59890d796d2c3009782c79e94d.pdf>
- Ghosh, S. (2018). How to I account for wrong spellings in chat bot. *Rasa Community Forum*. Retrieved from <https://forum.rasa.com/t/how-to-i-account-for-wrongspellings-in-chat-bot/1545>
- Grotz, B. (2017). *Grundkurs Linux* (0.2.7). Augsburg. Retrieved from [https://www.grundwissen.de/linux/\\_downloads/grundkurs-linux.pdf](https://www.grundwissen.de/linux/_downloads/grundkurs-linux.pdf)
- Helfferich, C. (2014). Leitfaden- und Experteninterviews. In *Handbuch Methoden der empirischen Sozialforschung* (pp. 559–574). Wiesbaden: Springer Fachmedien Wiesbaden. [https://doi.org/10.1007/978-3-531-18939-0\\_39](https://doi.org/10.1007/978-3-531-18939-0_39)
- Huttunen, H. (2015). *Modern Machine Learning in Python*. Tampere. Retrieved from <http://www.cs.tut.fi/~hehu/MMLP.pdf>

- Jacob, R., Heinz, A., & Décieux, J. P. P. (2013). *Umfrage - Einführung in die Methoden der Umfrageforschung*. München: Oldenbourg Wissenschaftsverlag.
- Jain, N., & Jain, A. (2017). A Survey on Popularity of Chat-Bots. *ICEMTE - International Conference On Emanations in Modern Technology and Engineering*, 5(3), 277–280. Retrieved from [http://www.ijritcc.org/download/conferences/ICEMTE\\_2017/Track\\_1\\_\(CSE\)/1488561500\\_03-03-2017.pdf](http://www.ijritcc.org/download/conferences/ICEMTE_2017/Track_1_(CSE)/1488561500_03-03-2017.pdf)
- Jenny, B. (2014). *Projektmanagement: Das Wissen für eine erfolgreiche Karriere* (4th ed.). Zürich: Vdf Hochschulverlag AG.
- Jindal, I. (2017). Building chatbots: Why message length matters. Retrieved March 11, 2019, from <https://www.invisionapp.com/inside-design/chatbots-messagelength/>
- Khademi, F., & Jamal, S. M. (2016). Predicting the 28 Days Compressive Strength of Concrete Using Artificial Neural Network. *I-Manager's Journal on Civil Engineering*, 6(2), 1. <https://doi.org/10.26634/jce.6.2.5936>
- Klopfenstein, L. C., Malatini, S., & Bogliolo, A. (2017). The rise of bots: A survey of conversational interfaces, patterns, and paradigms. *DIS '17 Proceedings of the 2017 Conference on Designing Interactive Systems*, 555–565. <https://doi.org/http://dx.doi.org/10.1145/3064663.3064672>
- Kollmann, T., Kuckertz, A., & Stöckmann, C. (2016). Ein Thema finden, eingrenzen und Ziele setzen. In *Das 1 x 1 des Wissenschaftlichen Arbeitens: Von der Idee bis zur Abgabe* (pp. 7–26). Wiesbaden: Springer Fachmedien Wiesbaden. [https://doi.org/10.1007/978-3-658-10707-9\\_2](https://doi.org/10.1007/978-3-658-10707-9_2)
- Kraus, R. (2009). Kraus, R. (2009). Agiles Projektmanagement - Voraussetzungen, Nutzen und Einsatzmöglichkeiten. In: IfM Impulse. Das Wissensmagazin des Instituts für Management. Ausgabe 3/2009, S. 4-13. *IfM Impulse*, 3, 4–13.
- Liddy, E. D. (2001). Natural Language Processing. In *In Encyclopedia of Library and Information Science* (2nd ed.). New York: Marcel Decker Inc. Retrieved from <https://surface.syr.edu/cnlp>
- McGrath, C. (2017). Chatbot Vocabulary: 10 Chatbot Terms You Need to Know. Retrieved March 6, 2019, from <https://chatbotmagazine.com/chatbotvocabulary-10-chatbot-terms-you-need-to-know-3911b1ef31b4>

- Microsoft Corporation. (2018). Prediction scores. Retrieved March 6, 2019, from <https://docs.microsoft.com/en-us/azure/cognitive-services/luis/luis-conceptprediction-score>
- Microsoft Corporation. (2019). Confidence Score. Retrieved March 6, 2019, from <https://docs.microsoft.com/en-us/azure/cognitiveservices/qnamaker/concepts/confidence-score>
- Misra, N. N. (n.d.). Anaconda versus Enthought Canopy? Retrieved February 2, 2019, from [https://www.researchgate.net/post/Anaconda\\_versus\\_Enthought\\_Canopy](https://www.researchgate.net/post/Anaconda_versus_Enthought_Canopy)
- Mornhinweg, N. (2018). Intelligente Chatbots im Gesundheitswesen – Prozessoptimierung im Gesundheitsbereich, von der Dienstleistung bis hin zum Controlling. In *Dienstleistungscontrolling in Gesundheitseinrichtungen* (pp. 143–160). <https://doi.org/10.1007/978-3-658-21710-5>
- Mortensen, P., & Zonca, A. (n.d.). Anaconda vs. EPD Enthought vs. manual installation of Python - Stack Overflow. Retrieved February 2, 2019, from <https://stackoverflow.com/questions/15762943/anaconda-vs-epd-enthought-vsmanual-installation-of-python>
- Muñoz-Romero, C., Ábalos, N., & Martín, J. (n.d.). ChatbotTest. Retrieved from <https://chatbottest.com/>
- Muñoz-Romero, C., Ábalos, N., & Martín, J. (2017). ChatbotTest. Retrieved March 9, 2019, from <https://github.com/chatbottest-com/guide/wiki>
- Neff, G., & Nagy, P. (2016). Talking to Bots: Symbiotic Agency and the Case of Tay. *International Journal of Communication*, 10(0), 17. Retrieved from <https://ijoc.org/index.php/ijoc/article/view/6277/1804>
- Niebert, K., & Gropengießer, H. (2014). Leitfadengestützte Interviews. In *Methoden in der naturwissenschaftsdidaktischen Forschung* (pp. 121–132). Berlin, Heidelberg: Springer Berlin Heidelberg. [https://doi.org/10.1007/978-3-642-37827-0\\_10](https://doi.org/10.1007/978-3-642-37827-0_10)
- Nimavat, K., & Champaneria, T. (2017). Chatbots: An overview. Types, Architecture, Tools and Future Possibilities. In *IJSRD - International Journal for Scientific Research & Development* (pp. 1019–1024). Ahmedabad. Retrieved from [https://www.researchgate.net/publication/320307269\\_Chatbots\\_An\\_overview\\_Types\\_Architecture\\_Tools\\_and\\_Future\\_Possibilities](https://www.researchgate.net/publication/320307269_Chatbots_An_overview_Types_Architecture_Tools_and_Future_Possibilities)
- Olivier, P. (2017). Introduction to NLP. Retrieved February 23, 2019, from <https://www.ekino.com/articles/introduction-to-nlp-part-i>

- Ovchinnikova, E. (2012). *Integration of World Knowledge for Natural Language Understanding* (1st ed., Vol. 3). Paris: Atlantis Press. <https://doi.org/10.2991/97894-91216-53-4>
- Panetta, K. (2016). Gartner's Top 10 Strategic Technology Trends for 2017. Retrieved January 22, 2019, from <https://www.gartner.com/smarterwithgartner/gartnerstop-10-technology-trends-2017/>
- Panetta, K. (2017). Gartner Top 10 Strategic Technology Trends for 2018. Retrieved January 22, 2019, from <https://www.gartner.com/smarterwithgartner/gartnerstop-10-strategic-technology-trends-for-2018/>
- Panetta, K. (2018). Gartner Top 10 Strategic Technology Trends for 2019. Retrieved January 22, 2019, from <https://www.gartner.com/smarterwithgartner/gartnerstop-10-strategic-technology-trends-for-2019/>
- Petraityte, J. (2018a). *Building a chatbot with Rasa NLU and Rasa Core on Vimeo*. Vimeo LCC. Retrieved from <https://vimeo.com/254777331>
- Petraityte, J. (2018b). Deprecating the state machine: building conversational AI with the Rasa stack. Retrieved March 10, 2019, from [https://www.slideshare.net/PyData/deprecating-the-state-machine-buildingconversational-ai-with-the-rasa-stack-justina-petraityt?from\\_action=save](https://www.slideshare.net/PyData/deprecating-the-state-machine-buildingconversational-ai-with-the-rasa-stack-justina-petraityt?from_action=save)
- Pröpper, N. (2012). *Agile Techniken für klassisches Projektmanagement - Qualifizierung zum PMI-ACP* (1st ed.). Heidelberg: mitp.
- Python Software Foundation. (2001). Python Software Foundation. Retrieved February 4, 2019, from <https://www.python.org/psf-landing/>
- Raj, S. (2019). *Building Chatbots with Python*. Building Chatbots with Python. Berkeley, CA: Apress. <https://doi.org/10.1007/978-1-4842-4096-0>
- Rao, A., & Verweij, G. (2017). *Sizing the prize: What's the real value of AI for your business and how can you capitalise?* Retrieved from <https://www.pwc.com/gx/en/issues/analytics/assets/pwc-ai-analysis-sizing-the-prize-report.pdf>
- Rasa Technologies GmbH. (2016). *rasa\_nlu*. Retrieved from [https://github.com/RasaHQ/rasa\\_nlu](https://github.com/RasaHQ/rasa_nlu)
- Rasa Technologies GmbH. (2018a). *Actions*. Retrieved March 3, 2019, from <https://rasa.com/docs/core/customactions/>

Rasa Technologies GmbH. (2018b). Chat & Voice platforms. Retrieved February 12, 2019, from <https://rasa.com/docs/core/connectors/>

Rasa Technologies GmbH. (2018c). Choosing a Rasa NLU Pipeline. Retrieved March 1, 2019, from [https://rasa.com/docs/nlu/choosing\\_pipeline/](https://rasa.com/docs/nlu/choosing_pipeline/)

Rasa Technologies GmbH. (2018d). Confidence and Fallback Intents. Retrieved March 6, 2019, from <https://rasa.com/docs/nlu/fallback/>

Rasa Technologies GmbH. (2018e). Debugging. Retrieved March 5, 2019, from <https://rasa.com/docs/core/debugging/>

Rasa Technologies GmbH. (2018f). Domain Format. Retrieved March 2, 2019, from <https://rasa.com/docs/core/domains/>

Rasa Technologies GmbH. (2018g). Fallback Actions. Retrieved March 10, 2019, from <https://rasa.com/docs/core/fallbacks/>

Rasa Technologies GmbH. (2018h). Getting Started with Rasa NLU. Retrieved January 26, 2019, from <https://rasa.com/docs/nlu/quickstart/>

Rasa Technologies GmbH. (2018i). How can I get training data for my chatbot? + more FAQs. Retrieved February 13, 2019, from <https://rasa.com/docs/faq/>

Rasa Technologies GmbH. (2018j). Interactive Learning. Retrieved March 3, 2019, from [https://rasa.com/docs/core/interactive\\_learning/](https://rasa.com/docs/core/interactive_learning/)

Rasa Technologies GmbH. (2018k). Language Support. Retrieved March 10, 2019, from <https://rasa.com/docs/nlu/languages/>

Rasa Technologies GmbH. (2018l). Migration Guide. Retrieved March 10, 2019, from <https://rasa.com/docs/core/migrations/>

Rasa Technologies GmbH. (2018m). Rasa NLU: Language Understanding for chatbots and AI assistants. Retrieved January 27, 2019, from <https://rasa.com/docs/nlu/>

Rasa Technologies GmbH. (2018n). Step 1: Understand the Rasa Stack. Retrieved January 27, 2019, from [https://rasa.com/docs/get\\_started\\_step1/](https://rasa.com/docs/get_started_step1/)

Rasa Technologies GmbH. (2018o). Step 2: Try it Out. Retrieved March 1, 2019, from [https://rasa.com/docs/get\\_started\\_step2/](https://rasa.com/docs/get_started_step2/)

- Rasa Technologies GmbH. (2018p). Story Data Format. Retrieved March 3, 2019, from <https://rasa.com/docs/core/stories/>
- Rasa Technologies GmbH. (2018q). Training Data Format. Retrieved February 10, 2019, from <https://rasa.com/docs/nlu/dataformat/>
- Rasa Technologies GmbH. (2019a). About Us. Retrieved March 11, 2019, from <https://rasa.com/about/>
- Rasa Technologies GmbH. (2019b). Rasa Platform. Retrieved January 28, 2019, from <https://rasa.com/products/rasa-platform/>
- Robinson, C. (n.d.). Brainwriting... this is a team exercise to develop new and creative ideas. Retrieved January 26, 2019, from <http://leadershipconsulting.com/brainwriting/>
- Russell-Rose, T. (2017). A Framework for Chatbot Evaluation - DZone Big Data. Retrieved March 8, 2019, from <https://dzone.com/articles/a-framework-for-chatbotevaluation>
- Russell, S., & Norvig, P. (2010). *Artificial Intelligence: A Modern Approach* (3rd ed.). New Jersey: Pearson Education, Inc. Retrieved from <https://faculty.psau.edu.sa/filedownload/doc-7-pdf/154ffbcec538a4161a406abf62f5b76-original.pdf>
- Sahin, M., Relieu, M., & Francillon, A. (2017). Using chatbots against voice spam: Analyzing Lenny's effectiveness. *SOUPS'17 Proceedings of the Thirteenth USENIX Conference on Usable Privacy and Security*, 319–337. Retrieved from <https://dl.acm.org/citation.cfm?id=3235950>
- SAS Institute Inc. (2019). Natural Language Processing - What it is and why it matters. Retrieved from [https://www.sas.com/en\\_nz/insights/analytics/what-is-naturallanguage-processing-nlp.html](https://www.sas.com/en_nz/insights/analytics/what-is-naturallanguage-processing-nlp.html)
- Sayadi, A. R., Monjezi, M., Talebi, N., & Khandelwal, M. (2013). A comparative study on the application of various artificial neural networks to simultaneous prediction of rock fragmentation and backbreak. *Journal of Rock Mechanics and Geotechnical Engineering*, 5(4), 318–324. <https://doi.org/10.1016/j.jrmge.2013.05.007>
- Scalableminds GmbH. (2019). Chatroom. Retrieved from <https://github.com/scalableminds/chatroom>



- Scherk, J., Pöchhacker-Tröscher, G., & Wagner, K. (2017). *Künstliche Intelligenz - Artificial Intelligence*. Linz. Retrieved from [https://www.bmvit.gv.at/innovation/downloads/kuenstliche\\_intelligenz.pdf](https://www.bmvit.gv.at/innovation/downloads/kuenstliche_intelligenz.pdf)
- Schulte, T., Zimmermann, D., & Müller, C. (2018). Einsatzmöglichkeiten von Chatbot-Interaktionen im schriftlichen Kundenservice. In S. Hess & H. Fischer (Eds.), *Mensch und Computer 2018 - Usability Professionals* (pp. 395–404). Bonn: Gesellschaft für Informatik e.V. Und German UPA e.V. Retrieved from <https://dl.gi.de/handle/20.500.12116/16783>
- Selle, S. (2018). *Künstliche Neuronale Netzwerke und Deep Learning*. Saarbrücken. Retrieved from [http://www.htwsaar.de/wiwi/fakultaet/personen/profile/sellestefan/Selle2018e\\_Kuenstliche\\_Neuronale\\_Netzwerke.pdf](http://www.htwsaar.de/wiwi/fakultaet/personen/profile/sellestefan/Selle2018e_Kuenstliche_Neuronale_Netzwerke.pdf)
- Shalev-Shwartz, S., & Ben-David, S. (2014). *Understanding Machine Learning*. Cambridge: Cambridge University Press. <https://doi.org/10.1017/CBO9781107298019>
- Shawar, B. A., & Atwell, E. (2007). Chatbots: Are they Really Useful? *LDV-Forum 2007*, 22(1), 29–49. Retrieved from <https://www.semanticscholar.org/paper/Chatbots%3A-Are-they-Really-UsefulShawar-Atwell/8d8284bfba7ebcb4e2575d864ec7c16ea6a168f0>
- Shieber, S. M. (1994). Lessons from a restricted Turing test. *Communications of the ACM*, 37(6), 70–78. <https://doi.org/10.1145/175208.175217>
- Shum, H.-Y., He, X., & Li, D. (2018). *From Eliza to Xiaolce: Challenges and Opportunities with Social Chatbots*. Retrieved from <https://arxiv.org/abs/1801.01957>
- Smart Platform Group. (2017). Articulate. Retrieved March 12, 2019, from <https://spg.ai/projects/articulate/>
- Statista GmbH. (2017a). Chatbots - Bevorzugte Persönlichkeit im Ländervergleich 2017. Retrieved February 2, 2019, from <https://de.statista.com/statistik/daten/studie/740032/umfrage/bevorzugtepersoenlichkeit-von-chatbots/>

- Statista GmbH. (2017b). Global chatbot market 2025. Retrieved February 3, 2019, from <https://www.statista.com/statistics/656596/worldwide-chatbot-market/>
- Statista GmbH. (2018). Global chatbot market 2024. Retrieved March 8, 2019, from <https://www.statista.com/statistics/966893/worldwide-chatbot-market-value/>
- Storp, M. (2002). Chatbots. Möglichkeiten und Grenzen der maschinellen Verarbeitung natürlicher Sprache. *NETWORX*, 25, 1–29. <https://doi.org/DOI.10.15488/2916>
- TensorFlow Community. (n.d.). Install TensorFlow with pip. Retrieved March 1, 2019, from <https://www.tensorflow.org/install/pip>
- The Apache Software Foundation. (2019). About the Apache HTTP Server Project - The Apache HTTP Server Project. Retrieved February 3, 2019, from [http://httpd.apache.org/ABOUT\\_APACHE.html](http://httpd.apache.org/ABOUT_APACHE.html)
- Traeger, M., Eberhart, A., Geldner, G., Morin, A. M., Putzke, C., Wulf, H., & Eberhart, L. H. J. (2003). Künstliche neuronale Netze. *Der Anaesthetist*, 52(11), 1055–1061. <https://doi.org/10.1007/s00101-003-0576-x>
- Treml, F. (2018a). Chatbot Training: Quest For the Holy Grail. Retrieved March 9, 2019, from <https://chatbotsmagazine.com/chatbot-training-quest-for-the-holy-grail6a028ca3a79b>
- Treml, F. (2018b). How to Test a Chatbot — Part 1: Why is it so hard ? Retrieved March 8, 2019, from <https://chatbotsmagazine.com/how-to-test-a-chatbot-part-1-whyis-it-so-hard-10f1ee8ca37d>
- Treml, F. (2018c). How to Test a Chatbot — Part 3: The Quest for Test Cases. Retrieved March 8, 2019, from <https://chatbotsmagazine.com/how-to-test-a-chatbot-part-3the-quest-for-test-cases-cb7a90238a40>
- Treml, F. (2018d). How to Test a Chatbot — Part 2: Superpowers for the testers. Retrieved March 8, 2019, from <https://chatbotsmagazine.com/how-to-test-achatbot-part-2-superpowers-for-the-testers-500cf6699b41>
- Turing, A. (1950). Computing Machinery and Intelligence. Retrieved February 24, 2019, from <https://www.abelard.org/turpap/turpap.php>
- Turkle, S. (1995). *Life on screen: Identity in the age of the Internet*. New York: Simon & Schuster.

- van Duin, S., & Bakhshi, N. (2018). *Artificial Intelligence*. Retrieved from <https://www2.deloitte.com/content/dam/Deloitte/nl/Documents/deloitteanalytics/deloitte-nl-data-analytics-artificial-intelligence-whitepaper-eng.pdf>
- VanGundy, A. B. (1984). Brainwriting for New Product Ideas: An Alternative to Brainstorming. *Journal of Consumer Marketing*, 1(2), 67–74. <https://doi.org/10.1108/eb008097>
- Wallace, R. S. (2009). The Anatomy of A.L.I.C.E. In *Parsing the Turing Test* (pp. 181–210). Dordrecht: Springer Netherlands. [https://doi.org/10.1007/978-1-4020-6710-5\\_13](https://doi.org/10.1007/978-1-4020-6710-5_13)
- Wang, X., Conboy, K., & Cawley, O. (2012). “Leagile” software development: An experience report analysis of the application of lean approaches in agile software development. *Journal of Systems and Software*, 85(6), 1287–1299. <https://doi.org/10.1016/J.JSS.2012.01.061>
- Weizenbaum, J., & Joseph. (1966). ELIZA — a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1), 36–45. <https://doi.org/10.1145/365153.365168>
- Williams, J. D., & Zweig, G. (2016). End-to-end LSTM-based dialog control optimized with supervised and reinforcement learning. Retrieved from <http://arxiv.org/abs/1606.01269>
- WordPress Foundation. (n.d.). REST API Handbook. Retrieved March 3, 2019, from <https://developer.wordpress.org/rest-api/>
- Wrobel, S., Joachims, T., & Morik, K. (2013). *Handbuch der Künstlichen Intelligenz*. (G. Görz, J. Schneeberger, & U. Schmid, Eds.), *Handbuch der Künstlichen Intelligenz* (Vol. 10.1524). München: DE GRUYTER. <https://doi.org/10.1524/9783486719796>
- Züger, R.-M. (2013). Sitzungen moderieren. In *Teamführung - Teams steuern und motivieren* (3rd ed.). Zürich: Compendio Bildungsmedien AG.

**What is your gender?**

- Female
- Male
- Prefer not to say

**What is the country of your home university?**

<List of countries>

**How long have you been or were you at The AI Institute?**

- One semester
- Two semester
- More than two semester

**Which degree programme are you visiting or have you visited at The AI Institute?**

<List of degree programmes>

**You are ...**

- ... an International student
- ... a double degree student
- ... an enrolled The AI Institute student

**How did you become aware of your studies at The AI Institute? (Multiple answers possible)**

- Home university (Partnership, advertising, guest lecturers, etc.)
- Fellow students of the home university
- The AI Institute's website
- Internet search engines (Google, Bing, etc.)
- Friends
- Other

**Did you have contact with the international office of The AI Institute before, during or after your stay at The AI Institute?**

- No
- Yes

**Could you give the reason(s) why did you got in touch with the international office? What kind of questions, problems or suggestions did you have?**

<Empty space for text>

**Did you have contact with the customer service / secretary's office of The AI Institute before, during or after your stay at The AI Institute?**

- No
- Yes

**Could you give the reason(s) why did you got in touch with the customer service / secretary's office? What kind of questions, problems or suggestions did you have?**  
<Empty space for text>

**Where did you got information on The AI Institute's school-related issues such as locations, study programmes, infrastructure, modules, etc.? (Multiple answers possible)**

- Home university  
 Fellow students of the home university  
 **The AI Institute's** website  
 **The AI Institute's** tutors  
 **The AI Institute's** international Office  
 **The AI Institute's** customer service / secretary's office  
 Other exchange students (via e-mail, WhatsApp, etc.)  
 Social media (Facebook, Instagram, Twitter, YouTube, etc.)  
 Other

**Where did you got information about private-related issues such as Paris, shopping, living, transport, parties, etc.? (Multiple answers possible)**

- Home university  
 Fellow students of the home university  
 **The AI Institute's** website  
 **The AI Institute's** tutors  
 **The AI Institute's** international Office  
 Other exchange students (via e-mail, WhatsApp, etc.)  
 Social media (Facebook, Instagram, Twitter, YouTube, etc.)  
 Internet search engines (Google, Bing, etc.)  
 Other

**Was there any information that was very difficult to find?**

- No  
 Yes

**What information was difficult to find?**

<Empty space for text>

**Did you had questions which were asked to The AI Institute but which were answered only partially, inaccurately, too late or not at all?**

- No  
 Yes

**Is there any information that you would have liked to know before your arrival in Hämeenlinna that would have helped you a lot with your preparation?**

- No  
 Yes

Appendix 1/3

**What information would you have liked to have had early enough?**

<Empty space for text>

***Explanation of a chatbot:*** A chatbot is a text-based dialogue system that allows chatting with a technical system. It has one area each for text input and output, which can be used in natural language to communicate with the system behind it (Wikipedia).

**Assuming a chatbot would be available on The AI Institute's website that could answer students' questions immediately, would you use it?**

- Yes  
 No

**Assuming a chatbot would be available on The AI Institute's website that could answer students' questions immediately, would you consider the chatbot as your first port of call for questions?**

- Yes  
 No

**Assuming a chatbot would be available on The AI Institute's website that could answer students' questions immediately, on which topics should he be able to give information? Please select the, in your opinion, three most important ones.**

- University and its infrastructure  
 Degree programmes and contact persons  
 Cafeteria and the food  
 Student accommodation  
 Town of Paris  
 Transport possibilities in Paris  
 Sports, leisure and travel activities  
 Shopping opportunities

**Assuming a chatbot would be available on The AI Institute's website that could answer students' questions immediately, which feature would be the most important to you?**

- Friendliness of the chatbot  
 Quality / Correctness of the chatbot's responses  
 Amount of knowledge of the chatbot  
 Availability of the chatbot

## INTERVIEW QUESTIONS

<b>How often are you contacted by people and institutions?</b>			
_____			
_____			
_____			
<b>Which channels are used to establish contact?</b>			
<input type="checkbox"/> E-mail	<input type="checkbox"/> Phone	<input type="checkbox"/> LinkedIn	<input type="checkbox"/> Facebook
<input type="checkbox"/> Letter	<input type="checkbox"/> Personal visit	<input type="checkbox"/> WhatsApp	<input type="checkbox"/> Instagram
<input type="checkbox"/> Other			
<b>What is the most used information channel?</b>			
_____			
_____			
_____			
<b>What are the reasons for contacting you?</b>			
<input type="checkbox"/> Requests for information <input type="checkbox"/> Complaints <input type="checkbox"/> Subsequent document delivery			
<input type="checkbox"/> Other			
<b>How many requests do you get on average per day / week / month?</b>			
_____			
_____			
_____			
<b>Are there certain times or periods at which more or less requests than usual are received?</b>			
_____			
_____			
_____			
<b>From whom are the requests?</b>			
<input type="checkbox"/> Press/Newspaper	<input type="checkbox"/> Active students	<input type="checkbox"/> Potential students	
<input type="checkbox"/> Researchers	<input type="checkbox"/> Universities	<input type="checkbox"/> Lecturers	
<input type="checkbox"/> Exchange students	<input type="checkbox"/> Job seekers	<input type="checkbox"/> Other	



## Appendix

<b>What kind of requests are there?</b>
<input type="checkbox"/> Questions about the study programmes <input type="checkbox"/> Questions about internships
<input type="checkbox"/> Questions about possible exchange semesters <input type="checkbox"/> Questions about tuition fees
<input type="checkbox"/> Questions about final/research/final projects <input type="checkbox"/> Questions about the campus
<input type="checkbox"/> Technical questions e.g. about the user account
<input type="checkbox"/> Questions about leisure time (traveling, skiing..)
<input type="checkbox"/> Questions about the status of something

3/2

<b>What is the course of the conversation, are there several back-and-forth actions or are most inquiries clarified with one or two answers?</b>
<hr/> <hr/> <hr/>
<b>On average, how long does it take to answer a request?</b>
<hr/> <hr/> <hr/>
<b>Do you think that the processing of requests could be faster? If yes, how?</b>
<hr/> <hr/> <hr/>
<b>How many employees do you have who take care of these inquiries?</b>
<hr/> <hr/> <hr/>
<b>How many hours a day do you estimate, are you and your department busy answering questions?</b>
<hr/> <hr/> <hr/>

## Appendix

<p><b>Are requests only processed during service hours?</b></p> <p>_____</p> <p>_____</p> <p>_____</p>
<p><b>Are inquiries also processed at weekends, during semester holidays, during lunchtime and breaks, or only during the official service hours?</b></p> <p>_____</p> <p>_____</p> <p>_____</p>
<p><b>Were there situations in which requests remained unanswered for a longer period of time? If yes, why?</b></p> <p>_____</p> <p>_____</p> <p>_____</p>
<p>_____</p>

3/3

<p><b>Is the availability of the requested information guaranteed? Is it possible to find the information they are requesting/looking for on the Internet or on your website?</b></p> <p>_____</p> <p>_____</p> <p>_____</p>
<p><b>Are there questions that cannot be answered without your help? Are there information that cannot be found on the website or on your social media platforms?</b></p> <p>_____</p> <p>_____</p> <p>_____</p>
<p><b>There were students who indicated that they had to contact The AI Institute because of the Learning Agreement. What exactly is their concern? How exactly does this procedure work?</b></p> <p>_____</p> <p>_____</p> <p>_____</p>
<p>_____</p>

## Appendix

**Would you be happy if frequently asked questions were reduced by the introduction of a chatbot?**

---



---



---

**Is there any information that you think the chatbot should be able to answer unconditionally (Are there, from your point of view, any very important things/answers that the future chatbot must have/know for sure)?**

---



---



---

**In answer to the question "Assuming a chatbot would be available on The AI Institute's website, which could answer questions of the students immediately, on which topics should he be able to give information? Please select the, in your opinion, three most important ones", I got the following results. Are you of the same mind?**

---



---



---

**In answer to the question "Was there any information that was very difficult to find? What information was difficult to find?", I got the following result. Is this information provided somewhere, or is it really missing?**

---



---



---

## Appendix

**In answer to the question "How did you become aware of your studies at The AI Institute?", I got the following result. Very many have indicated that there are partnerships between their university and The AI Institute. However, are there people who contact The AI Institute directly to negotiate an exchange semester or a double degree? \_\_\_\_\_**

**In answer to the question "Where did you get information on The AI Institute's school-related issues such as locations, study programmes, infrastructure, modules, etc.? (Multiple answers possible)?", I got the following result. Do you have any other places where you provide school-related stuff?**

## SYSTEM AND SOFTWARE INFORMATION

## Appendix 2

```
Hardware
=====

Running on           VMware Virtual Platform
Hard disk            34 GB, SCSI Hard disk
Memory               2 GB RAM
Swap                 2 GB HDD
CPU                  1
Architecture         x86_64
Model name           Intel(R) Xeon(R) CPU E5-2667 v4
                    @ 3.20GHz

Operating System
=====

Operating System     Linux Ubuntu 18.04.1 LTS
Kernel Version       4.15.0-43-generic

Software
=====

Anaconda              1.7.2
Apache2                2.4.29
Build-essential       12.4
Conda                  4.6.7
Git-core               2.17.1
Npm                    3.5.2
NodeJS                 8.10.0
Python                 3.6.4
Python-dev             2.7.15
```

PYTHON SCRIPT –

Appendix 3

INSTALLED PYTHON PACKAGES

### rasa_nlu gevent==1.3.7 klein==17.10.0 boto3==1.9.71 typing==3.6.6 future==0.17.1 six==1.12.0 jsonschema==2.6.0 matplotlib==2.2.3 requests==2.21.0 tqdm==4.28.1 numpy==1.14.5 simplejson==3.16.0
### Sklearn spacy==2.0.18 scikit-learn==0.19.2 sklearn- crfsuite==0.3.6
### rasa_core apscheduler==3.5.3 fakeredis==0.10.3 graphviz==0.10.1 h5py==2.8.0 jsonpickle==0.9.6 keras==2.2.4 pandoc==1.0.2 redis==2.10.6 tensorflow==1.10.0 networkx==2.2 ConfigArgParse==0.13.0 pykwalify==1.6.0 coloredlogs==10.0 ruamel.yaml==0.15.83 flask==1.0.2 rasa_nlu==0.13.8
### additional py pandoc==1.4 service-identity==18.1.0 rasa_core==0.12.3

PYTHON SCRIPT –  
NLU\_MODEL.PY

## Appendix 4

```
from rasa_nlu.training_data import load_data
from rasa_nlu.config import
RasaNLUModelConfig from rasa_nlu.model import
Trainer
from rasa_nlu.model import Metadata, Interpreter
from rasa_nlu import config
def
train_nlu():
    training_data = load_data('./data/data.json')
    trainer = Trainer(config.load('config_spacy.json'))
    trainer.train(training_data)
    model_directory = trainer.persist('./models/nlu' ,
fixed_model_name = 'hamknlu')
def run_nlu():
    interpreter =
Interpreter.load('./models/nlu/default/hamknlu')
print(interpreter.parse(u"How much does the
accommodation cost?"))
if __name__ ==
'__main__':
    #train_nlu()
    #run_nlu()
```

PYTHON SCRIPT –

Appendix 5

TRAIN\_INIT.PY

```
from __future__ import absolute_import
from __future__ import division
from __future__ import unicode_literals

import logging

from rasa_core.agent import Agent
from rasa_core.policies.keras_policy import KerasPolicy
from rasa_core.policies.memoization import
MemoizationPolicy
from rasa_core.policies.fallback import FallbackPolicy
if __name__ ==
'__main__':
    logging.basicConfig(level='INFO')

    training_data_file = './data/stories.md'
model_path = './models/dialogue'
fallback =
FallbackPolicy(fallback_action_name="action_default_fallba
ck", core_threshold=0.3, nlu_threshold=0.3)

    agent = Agent('hamk_domain.yml', policies =
[MemoizationPolicy(max_history=3),
KerasPolicy(max_history=3), fallback])

    data = agent.load_data(training_data_file)
agent.train(data)
agent.persist(model_path)
```



```
from __future__ import absolute_import
from __future__ import division
from __future__ import print_function
from __future__ import unicode_literals

import logging

from rasa_core.agent import Agent
from rasa_core.policies.keras_policy import KerasPolicy
from rasa_core.policies.memoization import
MemoizationPolicy
from rasa_core.interpreter import
RasaNLUIInterpreter from rasa_core.train import
interactive from rasa_core.utils import
EndpointConfig

logger = logging.getLogger(__name__)

def run_lisa_online(interpreter,
                    domain_file="hamk_domain.yml",
                    training_data_file='data/stories.md'):
    action_endpoint =
    EndpointConfig(url="http://localhost:5055/webhook")

    agent = Agent(domain_file,

    policies=[MemoizationPolicy(max_history=30),
    KerasPolicy(max_history=30, epochs=50, batch_size=150)],
    interpreter=interpreter,

    action_endpoint=action_endpoint)

    data = agent.load_data(training_data_file)
    agent.train(data)
    interactive.run_interactive_learning(agent,
    training_data_file)    return agent
    if __name__ ==
    '__main__':
        logging.basicConfig(level="INFO")
    nlu_interpreter =
    RasaNLUIInterpreter('./models/nlu/default/hamknlu')
    run_lisa_online(nlu_interpreter)
```

PYTHON SCRIPT –

DIALOGUE\_MGMT\_MODEL.PY

appendix 7

```

from __future__ import absolute_import
from __future__ import division
from __future__ import print_function
from __future__ import unicode_literals
import logging
import rasa_core

from rasa_core.run import serve_application
from rasa_core.agent import Agent
from rasa_core.policies.keras_policy import KerasPolicy
from rasa_core.policies.memoization import MemoizationPolicy
from rasa_core.interpreter import RasaNLUInterpreter
from rasa_core.train import interactive
from rasa_core.utils import EndpointConfig
from rasa_core.run import serve_application
from rasa_core import config

logger = logging.getLogger(__name__)
def train_dialogue(domain_file = 'hamk_domain.yml',
model_path = './models/dialogue', training_data_file
= './data/stories.md'):

    agent = Agent(domain_file, policies =
[MemoizationPolicy(), KerasPolicy(max_history=3,
epochs=300, batch_size=50)])
    data = agent.load_data(training_data_file)

    agent.train(data)

    agent.persist(model_path)
return agent
def
run_lisa_online(serve_forever=True):
    interpreter =
RasaNLUInterpreter('./models/nlu/default/hamknlu')
action_endpoint =
EndpointConfig(url="http://localhost:5055/webhook")
agent = Agent.load('./models/dialogue',
interpreter=interpreter, action_endpoint=action_endpoint)
rasa_core.run.serve_application(agent
,channel='cmdline')
return agent
if __name__ ==
'__main__':
    logging.basicConfig(level="INFO")
nlu_interpreter =
RasaNLUInterpreter('./models/nlu/default/hamknlu')
run_lisa_online(nlu_interpreter)

```