Automatic Generation of Legally and Ethically Correct Email Replies

by

Yannick Haveman & Steven Meijer

to obtain the degree of Bachelor of Science
at the Delft University of Technology,
to be defended publicly on July 5th, 2019

Project duration: April 22, 2019 – July 5, 2019
Supervised by Dr. J.S. Rellermeyer

The content of this thesis is highly confidential. Uncensored versions may only be viewed by the authors, client company and all its employees, TU Delft supervisor Dr. J.S. Rellermeyer, and TU Delft Bachelor's End Project coordinators Ir. O.W. Visser and H. Wang. Public versions may not become available until July 5, 2021. Even then, all publicly available versions shall be redacted to protect the integrity of the research, in addition to keeping the identity and details of the client private.

Delft
University of
Technology

Preface

This thesis was written as the final part of our Bachelor of Computer Science and Engineering at the TU Delft. In this work, we analyse how we can decrease the workload of our client company, [redacted], by implementing neural networks to perform named entity recognition (NER) and intent classification. This enables us to create a system that can easily extract important information from incoming emails and generate a reply to them. The research was performed over nine weeks and took place at the main office of [redacted], or at the office of [redacted]. This company is owned by Mr. Van Leeuwen, who has been responsible for the development of the system over the last two years. We would like to thank everybody at [redacted], our contact [redacted] in particular, in addition to Mr. van Leeuwen and our TU Delft coach Dr. Rellermeyer for their aid in creating this final product. Due to the sensitive nature of this subject, parts of this public paper are redacted or have been modified since it was graded, to allow us to show off the functionality that was implemented by us. The full, uncensored paper is available on request by sending an email to us, contact information is available in the infosheet attached to this paper. This modified version of the paper has been approved for upload by the Bachelor End Project Coordinators. Examples used in this paper might not be representable by actual data generated by the system. However, its functionality remains the same.

Yannick Haveman & Steven Meijer
Delft, June 2019

Table of Contents

1 Introduction

Artificial Intelligence (AI) is abundantly present in society nowadays, being implemented in websites and hardware by millions of companies. Up to 80% of all companies nowadays use some sort of AI (Bourne, 2018), mostly for the purpose of predictive analytics, machine learning and natural language processing (Rayome, 2018). For the latter use case, where AI is being used to correctly interpret human speech and languages, many implementations can be thought of. For example, phones use it to translate speech into text, translator applications use it to translate the meaning of a sentence into another language, and companies use chatbots to provide 24/7 customer support or otherwise automatically interact with customers or other Persons of Interest (POIs), to increase the efficiency of their system.

Most if not all chatbots are custom trained to perform a specific task in a certain field, as it can only respond to sentences that it has been trained on. For example, the possibilities of a chatbot could be demonstrated by the customer support chatbot of the Philips automatic train timetable information system (Aust, Oerder, Seide, & Steinbiss, 1995). In this system, an AI is capable of providing users with accurate and real-time data of most German train stations. Other implementations of chatbots can be found on nearly every web-shop, that extends their customer support with AI to assist customers with their questions.

Further applications of algorithms used for natural language processing include assisting law enforcement with their tasks. For example, when new laws are adopted, old criminal convictions should be expunged. With the help of scanned court files and automatic character recognition, a judge in California was able to clear thousands of records with the press of a single button (Lee, 2019). This algorithm determined whose criminal past could be expunged, based on keywords. The system disregarded any record that involved violent crime and automatically consulted a human if it was unsure of its findings. Additionally, the algorithm was able to fill in the required paperwork automatically.

There are more ways in which AI can assist law enforcement in their daily operations. However, due to the sensitivity of the subject and to protect the integrity of this research, the exact field of this paper will be redacted. Scattered across the web are many advertisements of[redacted] Often, these are harmless ads[redacted] Even worse, sometimes these advertisements are[redacted] (citation).

That is where the chat functionality of this paper comes in. Although many perfectly good implementations of chatbots and systems alike already exist, as noted before, they are trained for a specific purpose and their source code is generally not available to the public. For this reason, this paper will look into the development of a new system, code-named [redacted]. This chatbot is currently in development at [redacted], and functions as a basic email client which includes a basic response generator.

2
Research Objectives

We have been tasked to improve the system so it can actually be used in production. The first part of the research consists of recognising personal data including names, addresses, age and appearance of the POI. The system should then provide a database with the original email where all personal information is removed, to allow the anonymous data to be used in training the chatbot and other applications. The raw email with personal information should be stored separately, as it may be needed later to report the POI to the authorities.

The second part of the research consists of updating and improving the intent classification for incoming emails, as the current system is very basic and in its current form not accurate and human-like enough to actually be used. It is necessary for the bot to have an accurate classification and to be able to provide responses indistinguishable from a human, as [redacted] would like to have the system send replies with minimal human intervention. This will greatly reduce the workload on the few employees working with the system. In turn, this allows [redacted] to drastically upscale their operations to hopefully [redacted] The main question this research aims to answer reads:

> *"To what extent is it possible to create an autonomous, self-learning chatbot abiding by both legal and ethical constraints in the field of* [redacted]*, that is fed anonymised data from real-world scenarios?"*

This research question will be subdivided into smaller sub-questions, that will each be answered in a separate chapter. At first, background information on the current state of the system will be given, accompanied by a detailed explanation of how the company operates and how automation could significantly improve their effectiveness. In-depth explanations of the current situation will be given accompanied by the sub-questions:

> *"What is the current state of the system?"* and
> *"What is the usefulness of having automation in this field?"*

After explaining why it is necessary for this system to exist and what it currently looks like, this paper will discuss the legal constraints that the back-end and the response generator have to abide by. All the legal implications for the in-development system will be discussed, and legal motivation behind the choices made for the implementation of the chatbot will be provided by answering the research questions:

> *"What are the legal constraints for personal data collection and usage?"* and
> *"What are the legal constraints for a bot*[redacted]*?"*

Not only does this research have to take into account the legality of the bots' actions, but as [redacted] comes with additional ethical problems, those have to be taken into consideration too. This research will provide a comprehensive explanation and justification for the implementation of the system with regard to its ethical constraints by answering:

"What are the ethical considerations for saving personal data of even non-offenders?" and
"What are the ethical constraints for a bot [redacted]*?"*

The next chapter will discuss frameworks that could be relevant for this research and give details of the implemented system after improving it. This chapter will elaborate on what frameworks are best used in order to improve the system and motivation is given as to why some frameworks are preferred over others with similar capabilities. Then, details on the exact implementation of the improved system are given. This chapter is accompanied by the research questions:

"How can we assure that the gathered data can be used to train the system?",
"What frameworks could be used to enhance the abilities of the system?" and
"How are the chosen frameworks implemented to create the bots?"

After explaining how the system functions and the bots reason, this paper will dive deeper into the results of the research by evaluating problems and other obstacles we encountered during the project. We will evaluate the project as a whole and compare our final deliveries with the client requirements, which can be found in Appendix A.

Finally, this paper will look into possible future applications and developments in this field of research. We discuss how we envision the application being used when it is deployed, and how further improvements to the system can increase the efficiency of the workflow of [redacted] even more. This paper will conclude with definite answers to all research questions which will combine to formulate an answer to our main research question of the paper.

3 Background Information

In this section, the procedures of the client company will be explained, and the state of the system as it was delivered to us at the start of the project will be described in detail. Afterwards, global and local statistics on [redacted] will be given, to substantiate the need for this field of research. Finally, the issues [redacted] is currently facing will be analysed, which will demonstrate the need for improvement, hence the necessity of this research and the automation of the system. This chapter will then have answered the research questions: *"What is the current state of the system?"* and *"What is the usefulness of having automation in this field?"*

3.1 Modus Operandi of Client Company

The team of [redacted] places professionally made advertisements of [redacted] on websites like [redacted] and [redacted], where people can respond via email [redacted] An example of a previously used advertisement can be seen in Figure 3.1. Naturally, precautions have been taken to ensure the safe and legal usage of [redacted]

[redacted] and advertisements include red flags that should make one suspicious. In sections 4.5 and 5.4 of this paper, a more in-depth detailing will be given on these advertisements, as one might argue about the legality and morality of this [redacted]. Referring back to the example advertisement of Figure 3.1, the picture and text include multiple red flags:

1. [redacted]
2. [redacted]
3. [redacted]
4. [redacted]
5. [redacted]

The advertisement itself mentions [redacted] (citation). During the conversation, the operator that holds the conversation notifies the POI of actually [redacted] after which the operator continues to chat with the POI to see if they [redacted] [redacted] and therefore acting against the law under section [redacted] (citation).

The operator finally flags the POI for either [redacted].

[redacted].

[redacted]

Figure 3.1: Example advertisement

[redacted] after the POI is told [redacted] They are then either notified [redacted] or commended for [redacted] respectively, and told of the red flags that the advertisement contained.

Regardless of their willingness to [redacted] The questionable legality of this will be discussed in sections 4.2 and 4.3, but the short answer boils down to the rights that protect [redacted] outweigh the right of privacy. In some cases, a POI might [redacted]

Sometimes, though not often, [redacted] decides to file a report to the authorities for further investigation, so they can proceed to take legal actions the POI. They indicated a handful of filed reports over the last two years. Because of the time and effort it takes to draw up such a report, [redacted] often decides not to go through with the prosecution. Furthermore, when [redacted] decides to file a report to the police, they are not notified of the final verdict, making it hard to determine the effectiveness. Because of lack of time at the police force, they sometimes find other, larger cases to be of bigger importance, causing [redacted] reports to remain uninvestigated.

3.2 State of the Current System

At the start of this project, the system had minimal functionality and was in fact entirely unusable before multiple flaws were resolved. The system in place was configured in Python 3.5.2, whereas packages were automatically installed in a virtual environment using Buildout. The system was able to, after adding an email account to the database, read all emails from that account, and reply to those emails from within the Django web interface. The system that automatically keeps track of conversations (emails back and forth to the same POI on a certain advertisement) to organise them properly in the interface and back-end database was non-functional, but instead, every email was parsed as a new conversation.

Everyone who sent an email to the advertisement email address is automatically added as a POI. The advertisements and POIs are stored in the system, in addition to information about these POIs. Information includes names, age, location, profession and appearance, but these can only be inserted manually into the system. Finally, every POI has certain flags, like if they have already been told whether or not [redacted]

The AI capabilities of the system were limited to a basic TensorFlow with NLTK intent classification system, based off a tutorial (gk__, 2017). This system is used to attempt to recognise what the POI wants or asks, so an appropriate response can be sent back. Emails are split-up per sentence using punctuation and certain words often used to indicate a new sentence, as not every POI uses punctuation in their emails. They had a list of intents ranging from items like ”Meeting up tonight” and ”Meeting up tomorrow” to questions about price and pictures, and each sentence is categorised according to its most confident intent. Then, for each sentence, as long as the system is confident enough about its intent (they had a threshold of 0.8), the system picked a reply belonging to that intent at random and linearly stitched all replies together to form a reply email. Responses to intents were limited to no more than two per intent, and a lot of training sentences were duplicated across multiple intents and contained almost identical sentences, making the intent classification inaccurate and thus unusable in its current form.

3.3 System Overview

[redacted]

Figure 3.2: UML workflow diagram

The UML diagram in Figure 3.2 displays the system as explained in the previous section. It should be read from right to left. A MailEvent (which can contain zero or more attachments and zero or one flag that it is an email where [redacted] adds the email to the PersonThread of a Person (or POI, which is created if it doesn't exist yet). Additionally, every email belongs to a MailConnection, which in turn is also linked to the Account of the Person. An Account itself contains multiple variables, like a Site, [redacted]

3.4 Global Statistics on (subject)

Worldwide statistics show that [redacted] is a serious issue amongst humanity. As such, there are more independent organisations like [redacted] (citation), whose aim it is to [redacted] The personal mission of [redacted] is about more than [redacted], as they also emphasise on creating awareness amongst other people by publicly naming and shaming any criminals they help expose. Sentences given to criminals are public records by default, but [redacted] also publishes the sentences given to these criminals on their own social media pages such as Facebook and Instagram.

An accumulation of worldwide reports, which came forth from replicating and extending a previous meta-analysis (citation) with over 200 additional studies, show that [redacted] (citation). The aforementioned statistics only cover [redacted], but other types of [redacted] (citation).

[redacted] (citation). Other studies show evidence of [redacted] They also show that adults with a history of [redacted] (citation). A more in-depth analysis of these statistics, specifically concerning the ethical and economic concerns about [redacted], will be given in section 5.1.

3.5 Need for Automation

Over the past two years, a lot of data has been accumulated by [redacted] itself. All conversations have been held manually without the use of any artificial intelligence. The biggest factor for the number of responses an advertisement gets is the time the ad stays online. As respondents might report the advertisements once they find out [redacted] the website will take down the advertisement and will often block the email address of the user. Sometimes they will even block the IP that was used to make the account with. Creating a new email account, a new account on the advertising website and creating the advertisement itself all cost a significant amount of time. As automating this part of the process is nearly impossible, other fields have to be improved in order to improve the workflow of [redacted]. As many [redacted] there is a great need for automation in this field, so [redacted] can scale up their operations.

A single advertisement can get anywhere from a few to hundreds of replies. The biggest mailbox we had access to had over 600 incoming emails. [redacted] indicated that many of these emails are simply left untouched, as they do not have the resources to respond to each and every email. Currently, they do not have access to any tool or system to process all incoming emails, and as a result, many POIs and thus [redacted] are never investigated. Even with the in-development tool, they would still be forced to manually respond to all those emails. The AI system already in place can be classified as non-functional, as the replies it spits out are inaccurate and obviously generated by a machine.

Figure 3.3 shows the statistics for a small sample of advertisements where [redacted] employees were able to respond to all emails. It shows the number of respondents per advertisement, divided into three categories: people that did not finish the conversation at all, people who [redacted] and people who [redacted] As can be seen in the figure, most conversations are never finished, either because the [redacted] employee responded too late and the POI [redacted] or because they otherwise lost interest in the conversation. Notable advertisements are [redacted]

[redacted]

Figure 3.3: Respondent statistics to five example advertisements

Another issue that [redacted] is facing is the fact they want to have automatic entity recognition. They keep track of personal information like names, addresses, age and appearance of the POIs, in case they need to file a report to the authorities. They would like to have a system in place that can automatically scan emails and extract these entities from the text, as manually extracting them takes too much time. Additionally, these entities can then be removed from the text to completely anonymise it, so the text can be used for training purposes and other environments where personal data should not be tied to the original email.

3.6 Public Awareness

One of the aims of [redacted] is to create more public awareness of their cause. It is important to note that the personnel at [redacted] needs to be careful with whom they share personal or company information. As such, they do not publicly commercialise themselves. This does not mean that no information is shared at all, but any information that is brought out into the public will be done so under the name of the parent company of [redacted] whose name is often used when communicating to the public. [redacted] itself has multiple private meetings a month where they meet up with different stakeholders or the media. These parties are already more than aware of the procedures and legislative rules. The only place where [redacted] itself advertises their cause is at conferences where they may hand out informative pamphlets or flyers such as Figure 3.4.

[redacted]

Figure 3.4: Example awareness advertisement

3.7 Problem Anticipation

Possible problems we anticipate on is the lack of data, as [redacted] only has a few hundred conversations available for training. This is not enough data to train a neural network to its full potential. More so, the fact that most POIs do not use proper spelling and punctuation makes text analytics significantly harder. Additionally, many emails do not contain any useful information for the system at all, as some people simply reply with a variation on ”Hey when can you meet”. Nevertheless, by automating at least the first reply to POIs, we reduce the workload on [redacted] employees and significantly improve their effectiveness, as most POIs lose their interest if no reply is sent back to them within a short amount of time. Even an hour is sometimes too long for them, [redacted]

4

Legal Constraints

This field of research is bound by the law in many ways. The legality of the bots' actions itself, as well as the field of research as a whole, is seeking the edge of legal constraints. As such, these constraints have to be clarified and taken into account when implementing the bot, as to not nullify all the research that is put in. All of the actions taken by the bots have to be legally justifiable, and should always be within the boundaries of the law, albeit on its edge points. In this chapter, all the laws that this research has to deal with will be explained. With these explanations, the research questions: *”What are the legal constraints for personal data collection and usage?”* and *”What are the legal constraints for a bot* [redacted]*?”* will be answered. As this research focuses solely on Dutch cases, only the Dutch criminal laws will be covered.

4.1 Data Collection and Anonymisation

This research will target the collection of data at first. All of the incoming messages have to be anonymised so they can be used as training data in the future. Anonymisation does not only help with making the data more suitable for training by generalising it (Hermann et al., 2015), but it also makes sure the procedures are compliant with the latest privacy regulations captured in the General Data Protection Regulation (GDPR) (Council of European Union, 2016). Article 32 of the GDPR states that:

> *”Taking into account the state of the art, the costs of implementation and the nature, scope, context and purposes of processing as well as the risk of varying likelihood and severity for the rights and freedoms of natural persons, the controller and the processor shall implement appropriate technical and organisational measures to ensure a level of security appropriate to the risk”*

By pseudonymizing or completely anonymising the data and only using it for internal training, the POI does not have to be informed of their data being collected and stored. This ensures that the procedure of training the system is always GDPR compliant (Wess, 2017; Lubowicka, 2019). Of course, in order to be able to make a case out of the conversation if the POI has [redacted] the original, non-anonymised data has to be kept as well. In order to comply with regulations, [redacted] invokes GDPR article 23 section d, which states that organisations are exempted from the GDPR rights data subjects can invoke if they safeguard:

> *”[...] the prevention, investigation, detection or prosecution of criminal offences or the execution of criminal penalties, including the safeguarding against and the prevention of threats to public security;”*

Naturally, [redacted] still has to comply with article 32 and as such, it has to be ensured that all data is stored locally with sufficient encryption and other methods of data protection, to prevent anyone other than researchers and employees at [redacted] from having access to the data. How this will be ensured will be thoroughly explained in section 6.4.

4.2 Data Privacy

A POI does not break the law by replying to an advertisement if [redacted] This usually never happens, as the host of the website is responsible for upholding the law. However, we do still store their data in the database, as the right of [redacted] weighs more heavily than the privacy of a person. To make sure that any of the collected data does not fall into the wrong hands, [redacted] ensures it is stored with proper precautions as described in section 6.4. On top of that, all advertisements contain keywords and red flags like in Figure 3.1, which should have been a warning and should prevent people from responding to the advertisement in the first place. In the case that a POI [redacted] they will receive a message commending them for [redacted] as described in section 3.1

4.3 (subject)

Under the Dutch criminal law, article [redacted], it is illegal for any person to have any kind of [redacted] Moreover, it is even illegal to [redacted] even if this [redacted] As noted under article [redacted]:

> [redacted]

This is what [redacted] uses at their attempt to [redacted] as the POI is already punishable by law after [redacted] Although it was not allowed to use a [redacted] up until recently, in [redacted] a law was passed that allows for the usage of a [redacted] (citation). It is important for the bot to notify the POI in a natural and correct way that [redacted] As the advertisements placed on the websites mentioned before [redacted] Because the POI needs to [redacted] this notifying has to be done in a natural way, as to not give away the fact that they are actually talking to a bot or an operator at [redacted].

4.4 Luring versus Provoking

As noted in the previous chapter, the original advertisements used to lure the POI into starting a conversation, are made in such a way that [redacted] For legal reasons, all [redacted]

The advertisements themselves comply will all laws and regulations, but one can argue for the amount of provoking that is done during the conversation. The next chapter will argue for the morality of these practices.

The line between luring and provoking is thin, and it has to be ensured that the bot does not cross this line in any way. Otherwise, the research results might be nullified or in a worse scenario, the client company might be sued for [redacted] As any kind of provoking is illegal under criminal law section 47.1 sub 2, it has to be ensured that the bot will never reply with sentences that one might consider to be provoking. The bot does attempt to lure the POI into showing their real intentions by showing interest in them, [redacted] and naturally by placing the advertisement in the first place, which [redacted]

Dutch law does not mention the legality of using bait in order to lure the POI. The first and most notable verdict about the usage of bait by the police dates from 2008, where a judge ruled that the police did not act unlawfully by deploying a bait bike. It was ruled that the suspect was not provoked into doing something different from his initial intentions to steal the bike (Hoge Raad, 2008). Moreover, the authorities are known for deploying multiple kinds of bait, ranging from teenagers attempting to buy alcohol to see if they are stopped by the cashier, to setting up bait bikes, cars, homes, and even grandmothers, to see if someone is willing to steal from them (Cerberus, 2019). In other words, bait may not change the current situation and thereby entice people to act differently than they normally would. As the advertisements placed by [redacted] only add to the already enormous amount of [redacted] advertisements, nobody is provoked by these bait advertisements.

4.5 Fake Online Profiles

Likewise, Dutch law does not mention the legality of creating fake online profiles. [redacted] creates these fake online profiles using pictures of [redacted] As such, no identity fraud or other crime is committed by creating these online profiles and placing these [redacted] advertisements.

Websites such as [redacted], where these [redacted] advertisements are placed, do however have their own terms of service (TOS) which disallows [redacted] (citation). For this reason, advertisements that get reported by POIs [redacted] will immediately be removed from the website, and often the account associated with the ad will be terminated. For these websites, fake profiles are a huge and ever-increasing problem and identity deception is a big problem in the field of online social profiles in general (Tsikerdekis & Zeadally, 2015). Though it might be unethical, going against the TOS of these websites is a necessity, as it is the only way to [redacted]

5

Ethical Constraints

Just because something is not illegal by law, does not mean that there are no reasons to refrain from doing so. This field of study has a tremendous amount of ethical dilemmas to deal with. Everything from misusing [redacted] by deliberately violating their TOS, to creating fake online profiles accompanied by fake pictures of [redacted] to lure unsuspecting people into a trap, has moral constraints that have to be considered. This section will discuss all ethical implications of this field of study and the ethical constraints we and the system have to take into account. It is accompanied by the following research questions: *”What are the ethical considerations for saving personal data of even non-offenders?”* and *”What are the ethical constraints for a system* [redacted]*?”*

5.1 Psychological Damages

There is overwhelming evidence that [redacted] (citation). Moreover, the effects of [redacted] and as such, any person that [redacted] (citation).

There is great difficulty in determining if [redacted] as many other factors like [redacted] However, parents who reported [redacted] showed an increase in likeliness of engaging in [redacted] (citation). Furthermore, it should be noted that the primary data source of most studies with regard to this had been done through self-report measures, which are seen as far more methodologically sound than studies that use clinical reviews and/or agency records (citation).

Moreover, the damages also have an economic impact on society as can be seen from the data gathered by the Netherlands Mental Health Survey and Incidence Study. In this study, the economic impact of multiple types of [redacted] was further researched in [redacted] and further narrowed down into direct and indirect (non-)medical costs. Even though the exposure rate to [redacted] the annual cost of [redacted] was determined to be [redacted] (citation). As such, there is great importance in further preventive development in this field.

5.2 Violating TOS

As noted in section 4.5, by advertising on websites like [redacted] violates their TOS and causes them to take action against the fake account. Although it is not against the law to set up a fake profile as stated in the aforementioned section, deliberately violating the TOS of a website and thus abusing their system might pose ethical issues. As the issue of fake profiles on social sites is already a large and ever-growing matter (Vishwanath, 2018), [redacted] contributes to this problem by creating additional fake profiles.

And yes, indeed [redacted] adds to the number of fake profiles. However, after considering what the advantages are of creating these fake personas, one will quickly see that the benefits outweigh the downsides. As noted in the previous section, the psychological effects of having to deal with [redacted] will have a drastic impact on the rest of someone's life.

5.3 Language Usage by the POI

Naturally, the manner of speech on which a chatbot is trained determines the language the eventual model can recognise and classify. Like in the real world, the people responding to these [redacted] advertisements have a wide range of words they use and level of language they master. Usually, though, the language they use is exactly what you would expect, often being rather patronising or just outright vulgar. Under no circumstance, a [redacted] should have to receive [redacted] For example, a random example message that was in one of the email boxes reads:

> [redacted]

5.4 Fake Profiles of (subject)

It is important to note that all forms of communication happen through online personas. As such, people are more likely to respond differently than they would in real life. [redacted] might encourage this by [redacted] and whilst doing so, they might have to deal with [redacted]. The advertisements themselves are made to get the POI's to respond to them. Certain keywords as shown in Figure 3.1 are put into the advertisement, and although they are meant to draw attention, they should also immediately raise suspicion. Extensive research about the behaviour of [redacted] has been done to discover more about [redacted] (citation). On top of that, usage of the online world gives people an easy tool to behave differently and allows one to mask or anonymise themselves (Bullingham & Vasconcelos, 2013). This could affect people and encourage them to behave differently or worse than they would do so normally. In a way, [redacted] attempts to [redacted] the POI, as they [redacted] The definition of [redacted] is:

> [redacted]

It might be questionable how moral it is to [redacted] and the answer to the question, if it is ethically justified for the cause of [redacted] might differ per person. It can be said that it is generally accepted that for this cause, the means do in fact justify the ends. An example that shows the effectiveness of fake online profiles to [redacted] (citation).

6

Implementation Details

The entire system was built around the currently existing Django framework. As the implementation of TensorFlow with NLTK was outdated and barely functional, we built two separate systems from the ground up; one for named entity recognition and data anonymisation, and another one to replace the existing response generating functionality. In this section, both systems will be explained in detail, together with our thought process and why we chose certain frameworks over other frameworks with similar capabilities. The research questions accompanying this chapter read *"How can we assure that the gathered data can be used to train the system?"*, *"What frameworks could be used to enhance the abilities of the system?"* and *"How are the chosen frameworks implemented to create the bots?"*

6.1 Gathering Training Data

One of the issues that arose during the project is the acquisition of enough usable data. The data that was provided to us consisted of email conversations in the form of single responses and dialogues. Because of the limited amount of accounts and even more limited useful data in these accounts, we looked into gathering additional data from other sources. These include a download of Dutch Wikipedia articles (Wikipedia, 2019) and certain initiatives such as Pushshift's open data initiative (Baumgartner, 2019).

The emails that were provided to us are actual real-life examples to what the final product will have to work with, as they were taken from previous interactions with POIs. The biggest distinction between our data and the data that could be gathered from Wikipedia is that the context and grammar are far from the clean and neat writing scheme that Wikipedia adheres to. Moreover, underrepresented entities in the email data are infrequent in Wikipedia data as well, and using Wikipedia articles would thus greatly increase the size of our training data without adding much useful information. Additionally, for our dialogue bot, we needed more dialogues, something that is likewise a rare occurrence in Wikipedia articles. This meant we were unable to use data dumps from the Dutch Wikipedia as training data.

Pushshift is a big-date storage and analytics project. It is useful when you need to analyse or aggregate large quantities of data from a specific date range in the past, which are available in JSON object format. During the research phase of the project, they seemed like viable options to expand the amount of data available to us. However, for Pushshift's data initiative other problems arose. Even though there are a plethora of manners in which one can parse the data provided, the data itself lacks Dutch language availability. Furthermore, the textual data additionally lacks actual grammar context and it is therefore an equally unviable addition to our training data. Consequently, the only source of trustworthy and usable information, unfortunately, has to come from the responses that [redacted] received to prior advertisements. This does mean that the data-set used by our models is confined to this smaller set of data.

We looked further into getting location data from services like Google's Geocoding API (Google, 2019), but this is a paid service. Additionally. we soon found out it would be much easier to let a neural network decide upon location data, making the API unnecessary for our goals.

At first, we thought the limited amount of available data would result in a much less accurate model. However, with our final model, we were even able to recognise the underrepresented entities with proper accuracy. The performance of our entity recogniser is elaborated on in section 6.2. This also meant we did not have to research other time-consuming ways to gather more data, such as scraping twitter or other social media sites.

6.2 Named Entity Recognition and Data Anonymisation

Identifying named entities in a text has been researched for the past 18 years. Back in 1991, on the Seventh IEEE Conference on Artificial Intelligence Applications, one of the first research papers about named entity recognition was presented (Nadeau & Sekine, 2007, p. 4). It described a system that could recognise company names in a text by using heuristics and handcrafted rules. Starting in 1996, the research field of entity recognition accelerated and since then, numerous events have been dedicated to the field of automatic entity recognition. The first event to mention the naming of entities was MUC-6, in which they demonstrated being able to identify names of people, organisations and geographical locations in text based on low-level templates (Grishman & Sundheim, 1996).

Nowadays, named entity recognition is done based on syntax analysis by machine learning and is many times more accurate than in the 90s (Goyal, Gupta, & Kumar, 2018). Using the powerful hardware available today, almost every laptop can create a neural network without too much effort. As the field of entity recognition is starting to mature more, there are more than enough frameworks available for easy implementation of a NER. For this project, we opted for the usage of SpaCy's Named Entity Recognition.

6.2.1 SpaCy: Speed Comparison

SpaCy is generally commended for its speed in natural language processing tasks like tokenization, tagging, and word parsing. They provide easy to use functions combined with proper documentation for a pleasant user experience overall, especially for people that are new to the field of neural networks and language processing like ourselves. Speed comparisons between SpaCy, NLTK and other frameworks can be found in Table 6.1. Here, 100.000 plain-text documents were extracted from an SQLite database. They were processed with the NLP libraries listed to three additive levels of detail: tokenization, tagging, and parsing. Times are in milliseconds and report the time it took the pipeline to complete, excluding pre-processing times. It should be noted that these statistics were extracted from a system with an Intel I7-3770 from 2012, so performance today should be faster.

SYSTEM	TOKENIZE	TAG	PARSE
spaCy	0.2ms	1ms	19ms
CoreNLP	0.18ms	10ms	49ms
ZPar	1ms	8ms	850ms
NLTK	4ms	443ms	n/a

Table 6.1: Speed comparison between SpaCy, CoreNLP, ZPar, and NLTK. Source: `https://spacy.io/usage/facts-figures#benchmarks`

6.2.2 Recognisable Entities

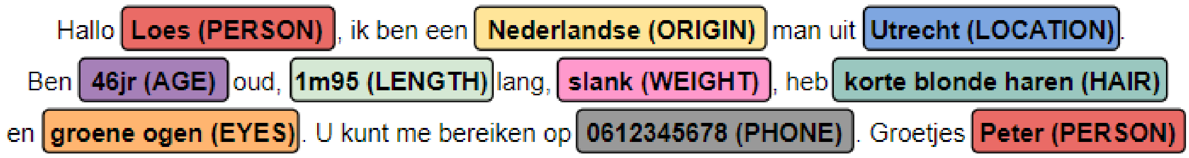
After deciding upon the framework to use, we had to download all emails from all 18 mailboxes we had access to. Mailboxes contained anywhere from a single to around thirty useful emails, that we could use for our entity recogniser. We downloaded the emails with the help of Google account download (all accounts were Gmail accounts) and extracted all email bodies. With the help of a tool created by

Murugavel (2019), we were able to quickly, though not flawlessly, annotate the entities we wanted. For this project, we chose to train the language processor with the following entities:

1. PERSON (Name)
2. LOCATION
3. AGE
4. LENGTH
5. WEIGHT
6. HAIR
7. EYES
8. PROFESSION
9. ORIGIN
10. PHONE

These properties were already in the database. Appearances such as ”LENGTH”, ”EYES” and ”ORIGIN” are stored as a combined field called ”LOOKS”, but in order to give the entity recogniser a better understanding of the different terms, we split them up. This also makes post-processing much easier. All entities are extracted from the email by the natural language processor, except for ”PHONE”. As phone numbers are written in many, predefined ways, we opted for extracting it with a regular expression instead.

As the entities are stored as a JSON object, duplicate entries are automatically removed. We debated on the exact approach we wanted to take with regard to entities like ”PERSON” being recognised multiple times in a text. An example can be seen in 6.1, where both the names of the POI and the advertising person are recognised. For now, we determined that it would be best to only keep the last entity it recognised, as this is usually the correct one.



Hallo Loes (PERSON), ik ben een Nederlandse (ORIGIN) man uit Utrecht (LOCATION). Ben 46jr (AGE) oud, 1m95 (LENGTH) lang, slank (WEIGHT), heb korte blonde haren (HAIR) en groene ogen (EYES). U kunt me bereiken op 0612345678 (PHONE). Groetjes Peter (PERSON)

Figure 6.1: Example messages showing ideal and perfect message annotations

6.2.3 NER Model Accuracy

As noted at the end of section 6.1, the final model is significantly more accurate than we initially thought. In total, we have around 260 sentences available to train on. For our final model, we used 200 annotated emails for training, and the remaining for testing. This approximately follows the generally accepted and long-standing standard of dividing the available samples 70%-30% for training and testing purposes respectively (Weiss & Indurkhya, 1993).

In total, we trained the model over 500 iterations and picked the model with the best F1 Score. In Figure 6.2, the F1 score per iteration can be seen. Precision and recall were generally not very divergent, and for our cause, one is not more important than the other. Precision is calculated with $TP/TP + FP$, recall with $TP/TP + FN$ and the F1 score by $2 * (Recall * Precision)/(Recall + Precision)$. Here, TP = True Positive, FP = False Positive and FN = False Negative. The best scoring model had a final loss of 43.27 and using the integrated scorer of SpaCy, we determined that our optimal model has the following scores:

1. Precision (ents_p): 82.96703296703298
2. Recall (ents_r): 79.47368421052632
3. F1 Score (ents_f): 81.18279569892472

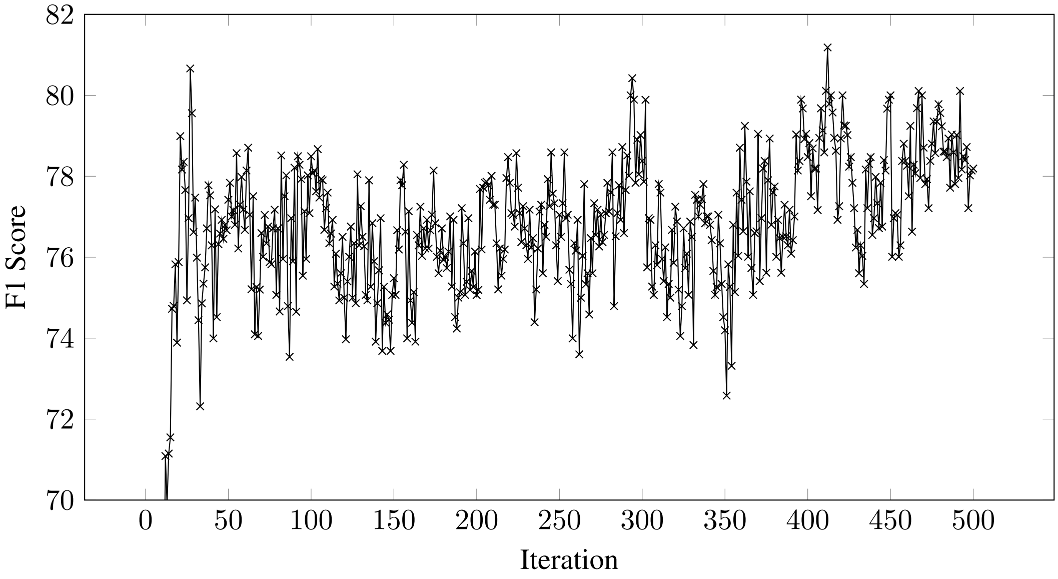


Figure 6.2: F1 Score at every iteration

6.3 Natural Language Understanding

To replace the currently existing system that performed intent recognition to automatically generate a reply, we looked into easy to use, but powerful systems. Besides more traditional implementations that use TensorFlow as a base, we looked into replacing the entire system with another framework. The latter is what we eventually went for, replacing the entire system with Rasa Natural Language Understanding, which is a relatively new tool for intent classification and entity extraction. This tool is very high level and thus makes it very easy to work with.

6.3.1 Researching Usable Frameworks

During the research period of this project, we discovered many frameworks that would bring us closer to a properly working solution. Frameworks we looked into included Theano, TensorFlow, Keras on top of either Theano or TensorFlow, and Rasa NLU, which also allows for different back-ends like Keras and SpaCy. The aforementioned libraries are all seen as good libraries to start working on deep learning applications.

- Theano is one of the first deep learning python libraries created. It has a great ability to handle the computations required for large neural networks. It is, however, difficult to understand for newcomers like us, and as such, other frameworks are more preferable to work with.

- TensorFlow is widely adopted by companies all over the globe. It supports desktop, mobile and embedded devices and has the advantage of being accessible from the Google Cloud platform and from the Amazon Web Services. The downsides of using TensorFlow is its learning curve, as it takes quite some skill to create a proper TensorFlow model.

- Keras is a deep learning library written in Python with the purpose of quick experimentation. It allows for the use of TensorFlow or Theano as back-end. Keras puts the user experience first and offers great modularity and extensibility in the form of new models. Albeit Keras looked promising and capable of handling the network we wanted to create, we instead chose Rasa NLU.

- Rasa NLU was deemed most appealing because of their customisability and so-called pipeline structure. A pipeline defines different components which process a message sequentially and leads to the classification of a message into intents. Their intent classification was a feature which we were looking for, as it gives us the ability to define the different intents and the actions per intent by hand. Furthermore, it allows for easy adaptability in the future.

6.3.2 Rasa Intent Classifications

Before our bot was capable of classifying an incoming message, we had to define the intents it could encounter. To be able to respond to as many different intents as possible, we decided to split up the sub-parts of the ”meeting_” intents into time of day. Also, to be able to distinguish between specific and general questions, we split up the ”question_possibilities” tag into two parts. [redacted] Our final model was trained on the following intents:

1. greet
2. meeting_now
3. meeting_today
4. meeting_today_morning
5. meeting_today_midday
6. meeting_today_evening
7. meeting_tomorrow
8. meeting_soon
9. meeting_future
10. question_appearance
11. question_age
12. question_price
13. question_location
14. question_transport
15. question_picture
16. question_possibilities_general
17. question_possibilities_specific
18. question_contact_info
19. own_appearance
20. [redacted]
21. [redacted]

With these intents, we believe the bot can reply to most messages it will encounter. However, if [redacted] feels the need to adjust or improve upon the existing model, they can easily do so by adding more intent classifiers that will be incorporated into the learning model.

6.3.3 Rasa Intent Responses

Likewise, the bot needed definitions of the actions it can take, that are linked to an encountered intent. As such, we additionally defined a list of actions the bot can use to reply to a message. We have chosen to incorporate the following actions:

1. utter_greet
2. utter_meeting_now
3. utter_meeting_today
4. utter_meeting_today_morning
5. utter_meeting_today_midday
6. utter_meeting_today_evening
7. utter_meeting_tomorrow
8. utter_meeting_soon
9. utter_meeting_future
10. utter_question_appearance
11. utter_age
12. utter_question_price
13. utter_question_location
14. utter_question_transport
15. utter_question_picture
16. utter_question_possibilities_general
17. utter_question_possibilities_specific
18. utter_question_contact_info

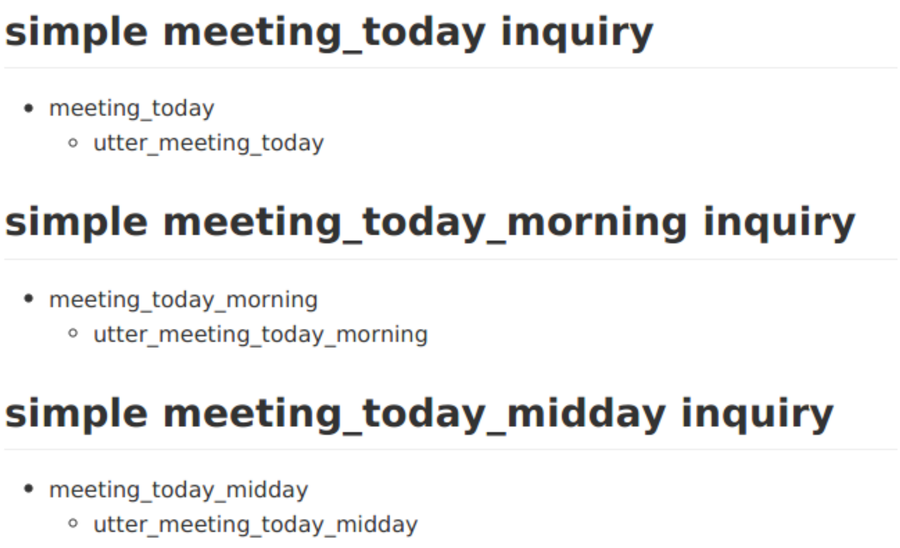
Even though most of the intents have an action, ”utter_age” differs slightly. It can be used to reply to a message asking about the age, or it is used to let the POI know the one they are talking to is [redacted]. Besides that, [redacted] are not included in the action list as they are only used to flag the POIs behaviour and do not necessarily require a response. The intent of ”own_appearance” is also ignored as it contains personal information of the POI, which does not require a response. Like the intent classifications, [redacted] can set or add responses to the system through the Rasa settings file.

Rasa does offer the possibility to combine different intents into a new intent and different actions into a new action. However, that would mean that we would need to create intents and actions for most of the combinations to ensure we can give answers to all the possible messages. This does not bode well for the usability and user-friendliness of the program and as such we have chosen for another approach. The message is split up into sentences using the following punctuation marks: {*.,:;!?*}. Then, for every sentence, we classify the intent, and for each intent found that meets the threshold of 0.7 we give a response that is randomly picked from a list containing possible answers. This is done so different replies do not look too similar. The system picks a random response from a provided list and can, for example, greet in the following manners:

1. Hey!
2. Hii,
3. Goeiedag,

6.3.4 Automatic Response Generation

To let the system know which of the available actions follow a certain intent, we filled out a document containing representations of actual dialogue, or stories, between a person and an AI system. Since we split the sentences into smaller parts we only had to create very basic stories, which only consist of an intent followed by an action. As such, every action is only connected to one intent which in turn simplifies the system. An example of this can be seen in Figure 6.3



simple meeting_today inquiry

- meeting_today
  - utter_meeting_today

simple meeting_today_morning inquiry

- meeting_today_morning
  - utter_meeting_today_morning

simple meeting_today_midday inquiry

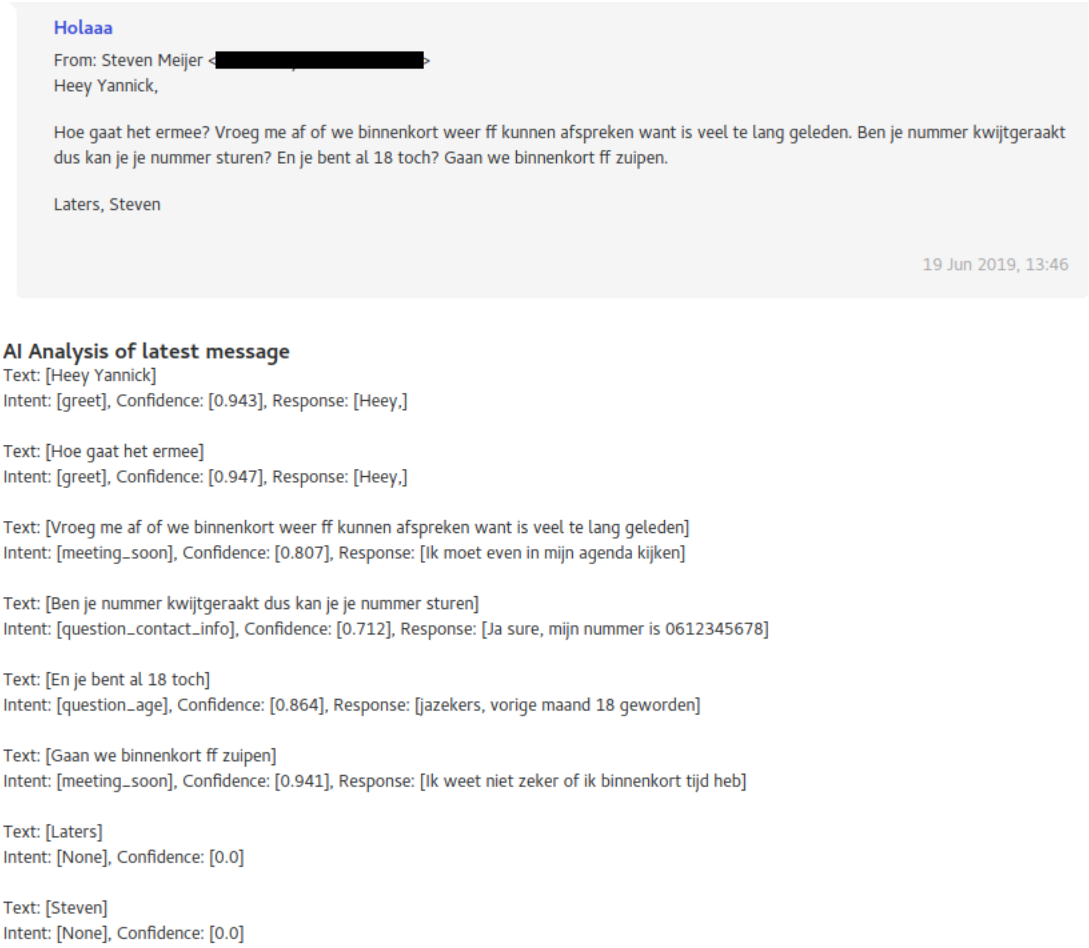
- meeting_today_midday
  - utter_meeting_today_midday

Figure 6.3: Example of matching the intents and actions through stories

Gathering and using the responses alone is not enough, as we might encounter duplicates which make the reply incoherent. Figure 6.4 shows how a received message is classified automatically. In this example, the system finds four different intents from the list in given subsection 6.3.2 and forms the responses for them automatically by matching them with the actions from the list in subsection 6.3.3. As the reply is being formed it is always ensured that a greet is at the start of the reply. The system appends the replies one by one whilst ignoring any duplicate it encounters. The final reply of the example given would be:

> *”Heey, Ik moet even in mijn agenda kijken. Ja sure, mijn nummer is 0612345678. Jazekers, vorige maand 18 geworden.”*

This final response handles all the questions given by a POI and can be sent out after manual review. Otherwise, it can be used as a template and an employee can manually edit any of the output before sending it. To ensure correct behaviour of the classification we have added test samples that verify the correctness of each intent. These tests contain anonymised or fictive data of which we know ourselves what the answer should be. With those tests we can verify the behaviour of each model is according to what we believe to be correct.



Holaaa

From: Steven Meijer <[redacted]>
Heey Yannick,

Hoe gaat het ermee? Vroeg me af of we binnenkort weer ff kunnen afspreken want is veel te lang geleden. Ben je nummer kwijtgeraakt dus kan je je nummer sturen? En je bent al 18 toch? Gaan we binnenkort ff zuipen.

Laters, Steven

19 Jun 2019, 13:46

AI Analysis of latest message

```
Text: [Heey Yannick]
Intent: [greet], Confidence: [0.943], Response: [Heey,]

Text: [Hoe gaat het ermee]
Intent: [greet], Confidence: [0.947], Response: [Heey,]

Text: [Vroeg me af of we binnenkort weer ff kunnen afspreken want is veel te lang geleden]
Intent: [meeting_soon], Confidence: [0.807], Response: [Ik moet even in mijn agenda kijken]

Text: [Ben je nummer kwijtgeraakt dus kan je je nummer sturen]
Intent: [question_contact_info], Confidence: [0.712], Response: [Ja sure, mijn nummer is 0612345678]

Text: [En je bent al 18 toch]
Intent: [question_age], Confidence: [0.864], Response: [jazekers, vorige maand 18 geworden]

Text: [Gaan we binnenkort ff zuipen]
Intent: [meeting_soon], Confidence: [0.941], Response: [Ik weet niet zeker of ik binnenkort tijd heb]

Text: [Laters]
Intent: [None], Confidence: [0.0]

Text: [Steven]
Intent: [None], Confidence: [0.0]
```

Figure 6.4: Example of an incoming email showing intent classification, the confidence and a response if the confidence level meets the threshold

6.4 Data Protection

This section is a follow-up on section 4.1 and elaborates how [redacted] and the system itself conforms to the law with regard to secure data storage and accessibility. As a first precaution to proper data protection, [redacted] requires each and every device that has access to any part of the system or documentation to be encrypted. This includes data transfers and hardware storage. On top of that, only a few selected people can access the most important parts of the system. Furthermore, if any of the stored data were to become corrupted or lost in any other way, encrypted data backups ensure the lost data can be recovered. As such, we had to ensure all of the data collected by the system is carefully protected to minimise risks.

Secure data storage is important, as the raw data will always be stored in the system. Both for legal uses when they file a report to the authorities, but also for the effective training of the bot prescribed in section 6.2, the system needs to have access to uncensored data. This real-world data is needed so the model can create an understanding of the entities listed in subsection 6.2.2. Only when the system is able to recognise all entities with enough confidence without lowering recall it will be possible for the system to work autonomously. Like with the outgoing replies, a human controller will still oversee the tokenization and entity recognition for the time being. This part of the system will never be sending out information to an external party, but it must still be ensured the information in the system is accurate.

To ensure all outgoing communication is both legally correct and human-like as to not scare off POIs, manual reviews of the generated responses are required to ensure the intended behaviour. The system as described in section 6.3 is responsible for all outgoing communication, which is generated based on the contextual information of incoming messages. This part of the system is only trained on data which is either fictional or completely void of personal details.

In conclusion, to ensure data collection happens in a legally responsible manner, we have taken extra precautions to ensure safe data collection and storage. Raw data is saved in a secure environment on encrypted hardware and we have implemented a system that can anonymise incoming email, so it can be used in further training and development.

7
Project Evaluation

After implementing the system, we can look back at our project and evaluate the process itself as well as the progress we were able to make in getting the system ready for deployment. This section will discuss all problems we encountered before and during the project, in addition to the solutions we implemented for those problems, or other workarounds we found. Finally, we discuss the state of the product itself and what client requirements we were and were not able to meet.

7.1 Product at the Start of the Project

As mentioned in section 3.2, the product as it was delivered to us at the start of the project was still in development, but had not been worked on for several months and was unfortunately barely functional overall. Additionally, as only a single person had maintained the system after initial development, documentation was not always present or up-to-date. Combined with the fact that is was not until the fifth week of the project that we obtained a working version of the system, the start of the project was slower than we hoped for.

7.2 Blocked Accounts and Changed Phone Numbers

As every fake profile has its own Google account with Two-Factor authentication enabled, we encountered issues when attempting to access these accounts. At first, we were told that we would need the SIM card associated with each account, which was all in a single binder somewhere in the office. As both our coordinator as the person in charge of the accounts were absent at the time, we had to wait until they came back before we could access the accounts. Fortunately, the next time we came into the office, the SIM cards seemed no longer needed, as Google recognised the IP address of the office. This also meant we did not have to deal with another issue of some SIM cards being assigned a new number, as they are from a cheap provider and had not been used for months. We downloaded all emails through Google's account download settings to be able to process them all offline.

7.3 NER Development

Once we had the system up and running, we were able to start the development of the AI system. We quickly found out that the current AI system based on NLTK was outdated and it was better to redesign the AI system from the ground up. In the research phase of the project, we came across the SpaCy Natural Language Processing framework, which looked like an excellent tool to use for our to be implemented Named Entity Recogniser. And indeed, the tool was very easy to use and proved its worth by providing us with excellent tools to create the NER we wanted.

The biggest obstacle in creating the NER was providing properly formatted training data to the model. SpaCy requires the entities to be annotated with their starting and ending character, and although we were able to use the annotator tool of Murugavel (2019), it still took us many days to get all the data we needed for a model that was reliable enough. Even then, there was only enough data available to train a proper model to recognise names, locations and ages with relatively good accuracy. Because we were able to create a regular expression to recognise phone numbers, our model is able to automatically fill those in as well. Other entities, mostly related to the appearance of the POI, are much harder to recognise as they are underrepresented in the training data. Although the system performs better than we initially thought it would, these entities are not recognised all the time. The ”PROFESSION” and ”ORIGIN” entities are only recognised occasionally, if at all, as the system simply does not contain enough data about those entities.

Despite the fact it would be possible to get more data or manually create representative data ourselves, this would be a very time-consuming activity. Instead, we decided to prioritise creating and optimising other parts of the system, as more training data will gradually become available through future advertisements anyway. We will ensure to train the employees at [redacted] so they are able to easily add more data to the training data already in place, so the system can keep improving over time.

7.4 NLU Development

For the development of the intent classification, we opted for Rasa NLU, as it is a recent and well-received framework for recognising the intent of a sentence. For now, we are using no custom tracker-store and thus are only able to handle single messages. Rasa comes with extended dialogue options to base output on previous input or variables, but due to time constraints we were unable to implement these functions. In the future, when [redacted] wishes to extend their operations[redacted] the system is expandable to take contextual information into account. This will, however, mean parts of the system will have to be rewritten.

We were surprised about the accuracy of Rasa, as even with a relatively low amount of sentences to train on, the system performs quite well, and is often able to provide good overall responses. Nevertheless, there is always room for improvement, so we made sure training of the system is documented and relatively easy, may they feel the need to do so. Although the intent classification is good overall, the replies the system comes up with are still quite disjoint and the replies themselves are simply direct rejections to the intent of the POI. As we are no professionals in generating these replies, we will leave the formatting of the actual replies to [redacted].

7.5 Client Contact and Communication

During the project, we were either present at the office of [redacted] or the office of iamIT. As both companies are very small, being limited to just a handful of people, we were easily able to get in touch with either our contact at [redacted] or with Michel, who knows most about the system. This came in handy plenty of times, as new problems arose during the project, or when we encountered new issues with the current system. They were always helpful and provided aid where needed, which was very useful to us for reasons explained in section 7.1. Communication during the project was clear overall, which made working on the project significantly more pleasing.

7.6 Final Deliveries

Overall, we believe we have managed to create a system that is viable enough to be used in production. Our system is able to identify most important information in a text, consisting of names, addresses and ages, and pass it on to the database. As the system is far from 100% accurate, getting the tokens still has to be done manually, so an employee can check if the system made the right decisions. Once the system does reach over 99% accuracy, it can be modified so human action is no longer a necessity, and only supervision is required.

Likewise, although the system is able to create proper intent classification and is able to generate a response whose content is related to the original message, this still has to be done manually, and no automatic emails are sent back yet. Because the responses generated by the system are easily modifiable, [redacted] is able to enter the exact reply they would like the system to generate.

At the start of the project, we received several client requirements, which we were supposed to integrate into the system. A detailed overview of the requirements can be found in Appendix A. Here, each requirement is listed first, with either a check mark indicating it is fully implemented, or a diamond indicating some work still needs to be done to ensure the requirement is fully met. Additionally, we give a detailed explanation for every requirement on how we integrated it into our system, or the reason why it has not been fully implemented. In total, we received six requirements, of which five are fully integrated into the system. [redacted]

8

Future Development

Although we were able to provide [redacted] with a functional and actually usable product, there are still improvements to be made. In this section, we will elaborate on all the future developments we believe would be beneficial for the system. Additionally, [redacted] disclosed their vision for the actual deployment of this system and changes that will have to be made to make it possible will be discussed here.

8.1 Obtaining More Training Data

As we have noted several times in this paper, the models as they currently exist are limited in their accuracy due to the lack of available training data. Gathering training data is always one of the biggest challenges of training neural networks, and our model is no exception. As [redacted] simply does not receive many emails that are useful for the system to train on, improving the system will be slow. Nevertheless, as operations will continue, more data will gradually become available, which can be annotated and then added to the system as training or test data. It should always be ensured the system uses around 30% of all available data for testing, so accuracy measurements are representative.

8.2 Improving NER and Implement Annotator Tool

To make the lives of employees easier, we would still like to implement a system similar to the annotator tool we used for this project (Murugavel, 2019), so new emails can easily be annotated and added to the system. This would make formatting new emails significantly more user-friendly, enabling even the employees with less technological knowledge to create new training data.

Besides improving the NER with more training data to make it more accurate, [redacted] already indicated they would like to have a system in place that can scan an entire [redacted], and extracts entities from it. Technically, this would already be possible with our system, as the system can scan entire text files for entities. To make it more accurate though, training data that consists of long paragraphs of text would have to be provided to the model. Additionally, some modifications have to be made, especially the management of entities being recognised multiple times in a text. As noted in subsection 6.2.2, we currently opted for the system to only store the latest entity it finds, as that is the correct entity in almost all cases we found. This would, of course, perform much worse if the input text gets longer, so a more robust system needs to be in place before entire cases can be reliably processed.

8.3 Improving Rasa Responses

Like mentioned in section 7.4, the replies the system currently comes up with are simple rejections of the request of the POI. This was done mostly to ensure that the system is properly working. Of course, we are no professionals and [redacted] themselves have experts in this field. As the replies Rasa picks from are simply picked from a settings file, the team of [redacted] can easily change the output of the system.

8.4 Upgrading Python Version

As the system currently relies on Python version 3.5.7, which is in ”security fixes only” mode, the system will have to be upgraded to a more up-to-date Python version in the near future. Additionally, Python 3.7 changes the way in which asynchronous methods work, making it easier to work with (Pranskevichus, 2019). Python 3.5.7 supports asynchronous methods but the implementation in 3.7 is many times more polished. Some Rasa methods like the training of the dialogue model require them being run asynchronously. Therefore, upgrading to Python 3.7 or higher will improve code readability and possibly functionality.

8.5 Switching Django System

The current system [redacted] is using for their operations is also based on Django, and some parts offer similar functionality as the in-development system we have been working on. Although they were still unsure exactly how to integrate the system we worked on with their workflow, they suggested that one of the options they are considering is extracting functionality from our system, and integrating it into their current system. This would mean that all functions we have worked on, in addition to other useful functions of our system, will have to be extracted and implemented into their system. This would be no problem as our code is isolated and well documented, but it might require some rewriting to make everything compatible.

9

Conclusions

The system as it is today is in a good and functional state. The methods created are isolated from the rest of the framework, which will allow for easy adaptability in the future. We have done our best to ensure that the data gathered and generated by the system is all stored separately, to ensure [redacted] always has all the data they need to create a case against a POI. The original, raw email data is kept safe in the event a lawsuit needs to be filed against the POI. Alongside the raw data, our anonymiser ensures an anonymous version of the data is available for testing and training purposes. This training data can be used for both the NER and Intent Classification capabilities of the system.

To ensure our choices would provide a proper, working solution that is easy to understand and maintain, multiple frameworks were researched. As the system is currently based on a Django framework, we do not foresee any problems in the future to being able to integrate functionality in a newer Django framework. Since [redacted] does not have enough manpower to do everything manually, we believe that the implemented AI models can be a useful aid for their operations.

We do note the system needs more training data for both the NER and Intent Classification part, in order to increase their accuracy. The NER should already be capable of correctly annotating messages with relatively high precision, which should help speed up the process of acquiring more training data, as new data only has to be checked. That, in turn, improves the automation process as a whole.

Overall, the system is not completely self-learning, but can easily be modified and abides by legal constraints by handling the data in a safe manner. Besides that, it adheres to ethical constraints with regard to[redacted] It is capable of learning from real-world scenarios and it has the ability to be retrained to increase performance in future iterations. It is not fully autonomous for now, but we envision a lot of possibilities to improve upon. Hopefully, one day in the future, the system will be able to work fully autonomously.

Returning to the main research question of the paper:

> *"To what extent is it possible to create an autonomous, self-learning chatbot abiding by both legal and ethical constraints in the field of* [redacted]*, that is fed anonymised data from real-world scenarios?"*

We conclude that although it is possible to create a good-enough chatbot that has additional functionality to analyse and extract named entities, it is not possible to create a system accurate enough to be run completely autonomous. Rules can be implemented to ensure legally and ethically correct operations, but human intervention will still be a necessity.

References

Aust, H., Oerder, M., Seide, F., & Steinbiss, V. (1995). The philips automatic train timetable information system. *Speech Communication*, *17*(3-4), 249–262.

Baumgartner, J. (2019, June 9). *Pushshift reddit comments.* Retrieved 2019-06-17, from `http://files.pushshift.io/reddit/comments/`

Bourne, V. (2018). *State of artificial intelligence for enterprises.* Teradata.

Bullingham, L., & Vasconcelos, A. C. (2013). 'the presentation of self in the online world': Goffman and the study of online identities. *Journal of information science*, *39*(1), 101–112.

Cerberus. (2019, June 5). *Stelende thuishulp betrapt door 'lokoma'.* Retrieved 2019-06-10, from `https://www.ad.nl/den-haag/stelende-thuishulp-betrapt-door-lokoma~a3dd713c/`

Council of European Union. (2016). *Council regulation (EU) no 2016/679.* (`https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32016R0679`)

gk_. (2017, May 7). *Contextual chatbots with tensorflow.* Retrieved 2019-06-10, from `https://chatbotsmagazine.com/contextual-chat-bots-with-tensorflow-4391749d0077`

Google. (2019). *Geocoding api.* Retrieved 2019-06-03, from `https://developers.google.com/maps/documentation/geocoding/intro`

Goyal, A., Gupta, V., & Kumar, M. (2018). Recent named entity recognition and classification techniques: a systematic review. *Computer Science Review*, *29*, 21–43.

Grishman, R., & Sundheim, B. (1996). Message understanding conference-6: A brief history. In *Coling 1996 volume 1: The 16th international conference on computational linguistics* (Vol. 1).

Hermann, K. M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., & Blunsom, P. (2015). Teaching machines to read and comprehend. In *Advances in neural information processing systems* (pp. 1693–1701).

Hoge Raad. (2008, October 28). *Ecli:nl:hr:2008:be9817.* (`https://uitspraken.rechtspraak.nl/inziendocument?id=ECLI:NL:HR:2008:BE9817`)

Lee, D. (2019, April 29). *An algorithm wipes clean the criminal pasts of thousands.* BBC News. Retrieved 2019-04-29, from `https://www.bbc.com/news/technology-48072164`

Lubowicka, K. (2019, 4). *The most important benefits of data pseudonymization and anonymization under gdpr.* Retrieved 2019-05-20, from `https://piwik.pro/blog/benefits-data-pseudonymization-anonymization-gdpr/`

Murugavel, M. (2019, April 22). *Spacy ner annotator.* Retrieved 2019-05-25, from `https://github.com/ManivannanMurugavel/spacy-ner-annotator`

Nadeau, D., & Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticae Investigationes*, *30*(1), 3–26.

Pranskevichus, E. (2019). *Optimized rule induction.* Retrieved 2019-06-24, from `https://docs.python.org/3/whatsnew/3.7.html`

Rayome, A. D. (2018, January 17). *61% of businesses have already implemented ai.* Retrieved 2019-04-29, from `https://www.techrepublic.com/article/61-of-businesses-have-already-implemented-ai/`

Tsikerdekis, M., & Zeadally, S. (2015). Detecting and preventing online identity deception in social networking services. *IEEE Internet Computing*, *19*(3), 41–49.

Vishwanath, A. (2018, September 20). *Why do so many people fall for fake profiles online?* Retrieved 2019-06-24, from http://theconversation.com/why-do-so-many-people-fall-for-fake-profiles-online-102754

Weiss, S. M., & Indurkhya, N. (1993). Optimized rule induction. *IEEE Expert*, *8*(6), 61–69.

Wess, M. (2017, 25). *Looking to comply with gdpr? here's a primer on anonymization and pseudonymization.* Retrieved 2019-05-20, from https://iapp.org/news/a/looking-to-comply-with-gdpr-heres-a-primer-on-anonymization-and-pseudonymization/

Wikipedia. (2019, April 29). *Wikipedia data.* Retrieved 2019-06-17, from https://en.wikipedia.org/wiki/Wikipedia:Database_download

18 references removed from public version

Appendices

A

Client Requirements

- ✓ The chatbot needs to have a changeable threshold: The client wanted an easy way to change the confidence threshold of the chatbot, after which it will be confident enough about the intent of the POI and generate a reply. We have implemented a settings file that allows for easy modification of all settings, including the fallback threshold for intent classification.

- ✓ The chatbot needs to be (re-)trained in an easy way: We implemented links in the interface that automatically train or re-train the NER, the NLU and the dialogue model. New training data can easily be added in a settings file.

- ✓ When the chatbot can not reply or understand a message, the moderator needs to be warned: When the threshold is not met for every sentence of the POI, the system falls back to a default answer. For now, this is simply a message telling the moderator that the system could not come up with a reliable response. As a moderator will continue to monitor all outgoing messages for now, we deem this to be a viable solution.

- ✓ The chatbot needs to be able to extract and mark important parts of the conversation for further actions: This two-fold client requirement consisted of first being able to mark important personal information in the conversation and store them. The second part of this requirement consists of finding out if the POI[redacted] Our system is able to reliably classify their intent, and sets a flag in the system which marks them as suspects or clears them of harm.

- ✓ Ability to anonymise data: The client wanted a way to automatically anonymise incoming emails, e.g. remove all personal information like names, addresses, age, and appearance. We implemented a system that can takes the entities is recognised, and filters them from the text, replacing them with a tag corresponding to the entity. For example, the age of the POI is changed to ”[AGE]”. Although we are reliably able to extract information people usually post about themselves like names, cities, and age, the system is struggling with appearance, due to lack of training data.

- ◇ The chatbot needs to notify the POI that they[redacted] We have a system in place that reliably classifies the intent of the POI. One of the intents is[redacted] No system is in place to notify the POI of[redacted]if the conversation has been going on for a while, but this could be implemented without too much effort. For [redacted], it is most important to be able to answer the first sentence anyway.

B
SIG Feedback

During the course of this project we had to send in our code to the *Software Intelligence Group*. They were able to provide us with feedback which we could use to improve our system. They tested our code based on multiple factors, and found three areas of improvement, which will be elaborated in the sections below. Our system scored a 3.6 out of 5 in their sustainability model, giving us a market average grade. SIG sent us their reply in Dutch, so their feedback will be in Dutch.

B.1 Unit Complexity

Unix complexity encapsulates the percentage of code that shows above average complexity, making it harder to understand and test. Their feedback:

> *Voor Unit Complexity wordt er gekeken naar het percentage code dat bovengemiddeld complex is. Dit betekent overigens niet noodzakelijkerwijs dat de functionaliteit zelf complex is: vaak ontstaat dit soort complexiteit per ongeluk omdat de methode te veel verantwoordelijkheden bevat, of doordat de implementatie van de logica onnodig complex is. Het opsplitsen van dit soort methodes in kleinere stukken zorgt ervoor dat elk onderdeel makkelijker te begrijpen, makkelijker te testen is, en daardoor eenvoudiger te onderhouden wordt. Door elk van de functionaliteiten onder te brengen in een aparte methode met een beschrijvende naam kan elk van de onderdelen apart getest worden, en wordt de overall flow van de methode makkelijker te begrijpen. Bij grote en complexe methodes kan dit gedaan worden door het probleem dat in de methode wordtd opgelost in deelproblemen te splitsen, en elk deelprobleem in een eigen methode onder te brengen. De oorspronkelijke methode kan vervolgens deze nieuwe methodes aanroepen, en de uitkomsten combineren tot het uiteindelijke resultaat. Voorbeelden in jullie project: - train ner.py:main()*

We ensured to split up methods we found to be too complex, including the method they mentioned. By making sure one function does not have too many responsibilities, the code not only becomes more readable, but also less prone to bugs.

B.2 Unit Size

Unit size encapsulates the percentage of code that is longer than average, again making the code harder to to understand and document. Their feedback:

> *Bij Unit Size wordt er gekeken naar het percentage code dat bovengemiddeld lang is. Dit kan verschillende redenen hebben, maar de meest voorkomende is dat een methode te veel functionaliteit bevat. Vaak was de methode oorspronkelijk kleiner, maar is deze in de loop van tijd steeds verder uitgebreid. De aanwezigheid van commentaar die stukken code van*

> *elkaar scheiden is meestal een indicator dat de methode meerdere verantwoordelijkheden bevat. Het opsplitsen van dit soort methodes zorgt er voor dat elke methode een duidelijke en specifieke functionele scope heeft. Daarnaast wordt de functionaliteit op deze manier vanzelf gedocumenteerd via methodenamen. Voorbeelden in jullie project: - train ner.py:main()*

We ensured to cut down on code length and keep all code well documented. By doing so, we ensure all people that read our code can understand what is going on without too much effort.

B.3 Unit Tests

The final improvement to our code is the availability of tests, as they ensure that the behaviour of the program remains according to the expected behaviour.

> *Als laatste nog de opmerking dat er geen (unit)test-code is gevonden in de code-upload. Het is sterk aan te raden om in ieder geval voor de belangrijkste delen van de functionaliteit automatische tests gedefinieerd te hebben om ervoor te zorgen dat eventuele aanpassingen niet voor ongewenst gedrag zorgen. Op lange termijn maakt de aanwezigheid van unit tests je code ook flexibeler, omdat aanpassingen kunnen worden doorgevoerd zonder de stabiliteit in gevaar te brengen.*

As such we have ensured that both the entity recognition and natural language understanding parts were tested. These tests verify that the end results are correct and that the program continues to work correctly with different versions and after training the bots with newly acquired data.

C
Infosheet

Before the end of the project, we were asked to create a small infosheet containing vital information about the project. This infosheet is one A4 and can be viewed on the next page.

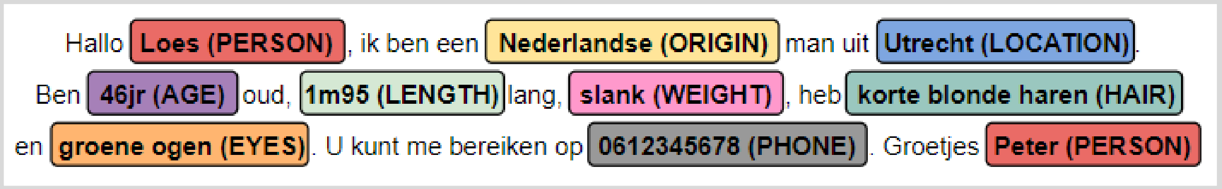
Automatic Generation of Legally and Ethically Correct Email Replies

Presentation Date: 05 July 2019

Description:

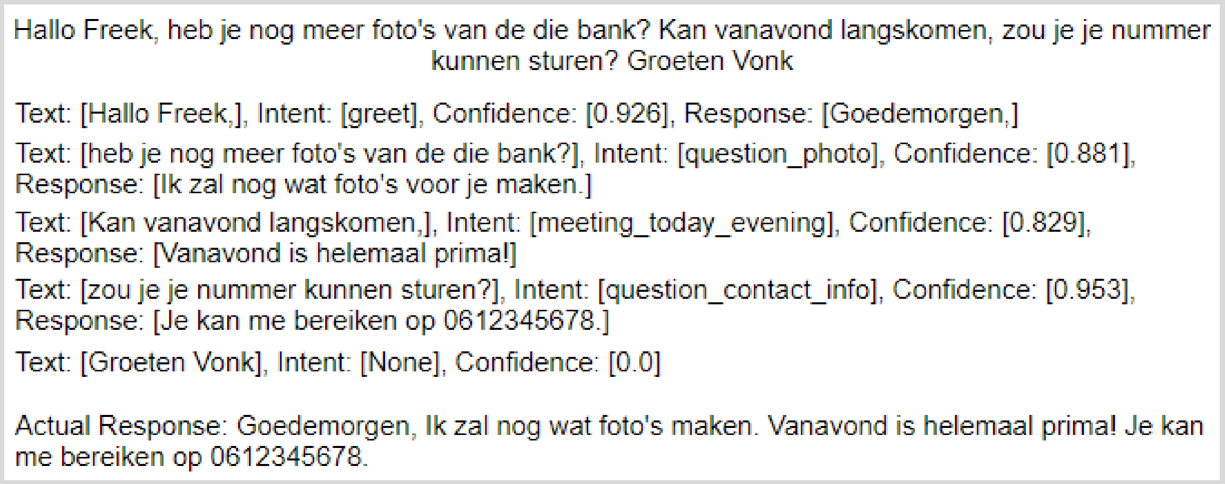
Even in 2019, customisable functions that perform advanced analysis on natural language are virtually non-existent. Models that are available to the general public are able to extract entities like someone's name and location, but these models are all trained for specific use-cases, and their accuracy is heavily dependent on the language used. Intent classification is likewise heavily dependent on context and language usage. Therefore, anyone who wishes to extract more than just basic information from messages, has to generate their own models, which are trained with their own custom training data.

For this research, many different frameworks were researched, but the final product uses SpaCy for named entity recognition and Rasa NLU for intent classification. Both systems combined, if properly configured with a set of rules, will provide an easy to use, robust and accurate system to generate email replies that are not only relevant, but also legally and ethically justified. The Named Entity Recogniser is able to distinguish between ten different entities, those being PERSON, AGE, LOCATION, HAIR, EYES, LENGTH, WEIGHT, PROFESSION, ORIGIN and PHONE. The intent classification system is able to split up entire emails and classify the intent per sentence. If the confidence threshold is met, a response is then formulated based on the intents found.



Hallo Loes (PERSON), ik ben een Nederlandse (ORIGIN) man uit Utrecht (LOCATION). Ben 46jr (AGE) oud, 1m95 (LENGTH) lang, slank (WEIGHT), heb korte blonde haren (HAIR) en groene ogen (EYES). U kunt me bereiken op 0612345678 (PHONE). Groetjes Peter (PERSON)

Fig. 1: Example of Named Entities



Hallo Freek, heb je nog meer foto's van de die bank? Kan vanavond langskomen, zou je je nummer kunnen sturen? Groeten Vonk

Text: [Hallo Freek,], Intent: [greet], Confidence: [0.926], Response: [Goedemorgen,]

Text: [heb je nog meer foto's van de die bank?], Intent: [question_photo], Confidence: [0.881], Response: [Ik zal nog wat foto's voor je maken.]

Text: [Kan vanavond langskomen,], Intent: [meeting_today_evening], Confidence: [0.829], Response: [Vanavond is helemaal prima!]

Text: [zou je je nummer kunnen sturen?], Intent: [question_contact_info], Confidence: [0.953], Response: [Je kan me bereiken op 0612345678.]

Text: [Groeten Vonk], Intent: [None], Confidence: [0.0]

Actual Response: Goedemorgen, Ik zal nog wat foto's maken. Vanavond is helemaal prima! Je kan me bereiken op 0612345678.

Fig. 2: Example of intent classification and response generation

Members of the project team:

Yannick Haveman: yanhave@gmail.com
Steven Meijer: stevenmeijer9@gmail.com

Coach and client:

Dr. J.S. Rellermeyer: TU Delft supervisor

As the client requests to remain anonymous, client information can be obtained through the members of this project

The final report for this project can be found at: **http://repository.tudelft.nl**