A deep learning approach to natural language generation for emotional agents

Author: Ridvan Aydin SIBIC
Student number: H00286088

Supervisor: Prof. Oliver LEMON
Second reader: Dr. Murdoch James GABBAY

A thesis submitted in fulfilment of the requirements for the degree of MSc.
in the
School of Mathematical and Computer Sciences

August 2018
Declaration of Authorship

I, Ridvan Aydin Sibic, declare that this thesis titled, 'A deep learning approach to natural language generation for emotional agents' and the work presented in it is my own. I confirm that this work submitted for assessment is my own and is expressed in my own words. Any uses made within it of the works of other authors in any form (e.g., ideas, equations, figures, text, tables, programs) are properly acknowledged at any point of their use. A list of the references employed is included.

Signed: Ridvan Aydin Sibic

Date: 15/08/2018
Abstract

Modern day dialogue generation for any type of artificial agents requires a lot of time and resources. Especially the generation of parallel corpora for simulating different emotions with the same core meaning requires a large time investment.

An emotion can be expressed with the use of certain phrases or words to make an utterance for example seem more happy or angry. The generation of dialogues with emotional differences is important in the creation of believable and more engaging agents in all dialogue based artificial systems.

A more automated way of producing dialogues with the same meaning but different emotional tendencies would greatly reduce the cost involved in manually writing the different tendencies. This project focuses on using a deep learning model to transform a given utterance into a desired sentiment or emotion.

The model has been trained and implemented into a dialogue based video game. The game was then played and evaluated by a number of participants.

The results of the evaluation show that some emotions are easier to recognize than others, with angry being an emotion that was comparatively easy for users to correctly identify, while the emotion happy was not. Another conclusion from the evaluation was that users did not conclusively prefer the emotional agents over the neutral agent.
Acknowledgements

I would like to thank my supervisor Prof. Oliver Lemon for his help and guidance without whom this research paper would not have been possible. Additional thanks to Ioannis Papaioannou for pointing me toward interesting and useful technologies.
Contents

Declaration of Authorship i

Abstract ii

Acknowledgements iii

Contents iv

List of Figures vii

List of Tables viii

Abbreviations ix

1 Introduction 1
   1.1 Motivation ................................................. 1
   1.2 Objectives ............................................... 2

2 Literature Review 4
   2.1 Emotional Models ........................................ 4
      2.1.1 OCC ..................................................... 4
      2.1.1.1 Classification ...................................... 5
      2.1.1.2 Quantification ...................................... 6
      2.1.1.3 Interaction .......................................... 7
      2.1.1.4 Mapping ............................................. 8
      2.1.1.5 Expression .......................................... 7
      2.1.1.6 Conclusion .......................................... 8
   2.2 Personality Models ....................................... 8
   2.3 Dialogue management ...................................... 9
      2.3.1 Rule based dialogues .................................. 9
      2.3.2 Dialogue trees ......................................... 10
   2.4 Style transfer ........................................... 10
      2.4.1 Sequence-to-Sequence systems ......................... 10
   2.5 Dataset ................................................... 11
      2.5.1 Sentiment dataset ..................................... 12
      2.5.2 Creating datasets ..................................... 12
# Contents

2.6 Speech synthesis .................................................. 13
2.7 Related Work ....................................................... 13
  2.7.1 Style transfer .................................................. 13
    2.7.1.1 Delete, Retrieve, Generate .......................... 13
    2.7.1.2 Polite Dialogue Generation Without Parallel Data .... 14
    2.7.1.3 Style Transfer Through Back-Translation ............ 15
  2.7.2 Emotionally expressive agents ............................ 15
    2.7.2.1 Façade: Handcrafted Emotional Output ............. 16
    2.7.2.2 ERiSA: Using the OCEAN model .................... 17
    2.7.2.3 ARIA VALUSPA: Extending ERiSA ................. 19
    2.7.2.4 FAtiMA .................................................. 20
  2.7.3 Evaluation of emotional output systems .................. 21
2.8 Summary .......................................................... 22

3 Methodology ......................................................... 23
  3.1 Data collection ................................................ 23
  3.2 Emotional NLG .................................................. 23
  3.3 Model integration into testing environment ................ 24
  3.4 Experiment ..................................................... 24
  3.5 Results ........................................................ 24

4 Implementation ..................................................... 25
  4.1 Dataset ......................................................... 25
    4.1.1 Data form ................................................ 26
    4.1.2 Gathered data .......................................... 26
  4.2 Sequence-to-Sequence model ................................. 27
    4.2.1 Design ................................................... 27
    4.2.2 Embedding ................................................. 28
    4.2.3 Recurrent unit .......................................... 29
      4.2.3.1 Long-Short-Term-Memory .......................... 30
      4.2.3.2 Gated-Recurrent-Unit ............................ 30
  4.2.4 Training results .......................................... 30
    4.2.4.1 Training speed ..................................... 31
    4.2.4.2 Training and validation ............................ 31
  4.3 Game Implementation ......................................... 35
    4.3.1 Initial game ............................................. 35
    4.3.2 Game adjustments ....................................... 37
      4.3.2.1 Game setting ...................................... 37
      4.3.2.2 Game dialogue ................................... 38

5 Legal, Ethical and Professional issues .......................... 40
  5.1 Legal issues ................................................... 40
  5.2 Ethical issues ................................................ 40
  5.3 Professional issues .......................................... 41

6 Evaluation and Results ........................................... 42
  6.1 Experiment design ............................................. 42
    6.1.1 The "before" questionnaire ............................ 43
## Contents

6.1.2 The "after" questionnaire ................................................. 43
6.1.3 Game experiment .......................................................... 44
6.2 Results .............................................................................. 44
6.3 Hypothesis testing .............................................................. 47
  6.3.1 Hypothesis I: Users correctly identified the angry agent as more angry than the neutral one. ........................................ 47
  6.3.2 Hypothesis II: Users correctly identified the happy agent as more happy than the neutral one. .................................... 48
  6.3.3 Hypothesis III: Users enjoy interacting with an angry agents more than with neutral ones. ............................................. 49
  6.3.4 Hypothesis IV: Users enjoy interacting with a happy agents more than with neutral ones. ............................................. 49

7 Conclusion and Future Work .................................................. 50
  7.1 Conclusion ....................................................................... 50
  7.2 Future Work ..................................................................... 51

A Consent form ........................................................................ 53
B Instructions ........................................................................... 55
C Questionnaire "before" ............................................................. 57
D Questionnaire "after" .............................................................. 60
E Data gathering form ............................................................... 64
Bibliography ............................................................................. 67
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>The original OCC model [Bartneck, 2002]</td>
</tr>
<tr>
<td>2.2</td>
<td>A basic dialogue interaction architecture [Aria-agent.eu, 2015]</td>
</tr>
<tr>
<td>2.3</td>
<td>Sequence-to-Sequence model [Sutskever et al., 2014]</td>
</tr>
<tr>
<td>2.4</td>
<td>Façade interactive drama dialogue example [Mateas and Stern, 2003]</td>
</tr>
<tr>
<td>2.5</td>
<td>Façade interactive drama architecture [Mateas and Stern, 2003]</td>
</tr>
<tr>
<td>2.6</td>
<td>Game flow. Action streams are selected based on the social relation value $R$ which is either 'very close', 'close'. 'Intro' is used when the user is unknown [Chowanda et al., 2014]</td>
</tr>
<tr>
<td>2.7</td>
<td>ARIA framework architecture, composed of modular Input, Agent Core, and Output blocks [Valstar et al., 2016]</td>
</tr>
<tr>
<td>2.8</td>
<td>FAiMA Core Architecture [Dias et al., 2014]</td>
</tr>
<tr>
<td>2.9</td>
<td>Smile Simulation. The blue and purple lines show the character’s urge to smile for Poppy and Spike, respectively. The orange line indicates the threshold to display a small smile and the purple line the threshold for smiling uncontrollably [Chowanda et al., 2014]</td>
</tr>
<tr>
<td>4.1</td>
<td>GRU character based design</td>
</tr>
<tr>
<td>4.2</td>
<td>The different RNN unit types, from left to right, standard RNN, LSTM and GRU [Olah, 2018]</td>
</tr>
<tr>
<td>4.3</td>
<td>The training and validation results graphs of the word embedded model</td>
</tr>
<tr>
<td>4.4</td>
<td>The training and validation results graphs of the character embedded model</td>
</tr>
<tr>
<td>4.5</td>
<td>The Speech-Driven Game System system architecture [Agnus et al., 2018]</td>
</tr>
<tr>
<td>4.6</td>
<td>A conversation in the game.</td>
</tr>
<tr>
<td>4.7</td>
<td>The games dialogue tree.</td>
</tr>
<tr>
<td>6.1</td>
<td>The mean values on how the participants felt about the game mode</td>
</tr>
<tr>
<td>6.2</td>
<td>The mean values on how the participants felt about the agent</td>
</tr>
</tbody>
</table>
List of Tables

4.1 Processing speed of one epoch ........................................ 31
4.2 Models trained on the angry dataset (0 = no match, 1 = sentiment or
   semantic match, 2 = sentiment and semantic match) ............... 35
4.3 Models trained on the happy dataset (0 = no match, 1 = sentiment or
   semantic match, 2 = sentiment and semantic match) ............... 35
4.4 Sample one output of the models ..................................... 36
4.5 Sample two output of the models ..................................... 36

6.1 McNemar test on Hypothesis I ....................................... 47
6.2 McNemar test on Hypothesis II ..................................... 48
6.3 McNemar test on Hypothesis III .................................... 49
6.4 McNemar test on Hypothesis IV .................................... 49
### Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>NLP</td>
<td>Natural Language Processing</td>
</tr>
<tr>
<td>NPC</td>
<td>Non-Player Character</td>
</tr>
<tr>
<td>STT</td>
<td>Speech-To-Text</td>
</tr>
<tr>
<td>TTS</td>
<td>Text-To-Speech</td>
</tr>
<tr>
<td>RNN</td>
<td>Recurrent-Neural-Network</td>
</tr>
<tr>
<td>LSTM</td>
<td>Long-Short-Term-Memory</td>
</tr>
<tr>
<td>GRU</td>
<td>Gated-Recurent-Unit</td>
</tr>
<tr>
<td>seq2seq</td>
<td>sequence-to-sequence</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Motivation

The need for emotionally expressive agents is constantly rising. Especially in the video game industry there is a constant need to improve dialogues and the tools for creating dialogues, because of the ever growing consumer expectations and game world sizes. This leads to many companies spending large sums and time on dialogue creation in games (e.g. GTA 5 and Skyrim). But even with an ever growing budget the creation of hand craft parallel dialogues for every possible emotional state might not be cost efficient as some dialogues might never be seen by used. The more divers the dialogue options get the fewer people will see each dialogue. Developer sometimes create the illusion of a lively world by limiting the number of lines in a dialogue and just cloning them onto multiple characters. But this illusion is broken as soon as the player tries to interact with these characters and only gets back the same three lines over and over again [Kerr and Szafron, 2009].

We therefore need a more automated solution, one that can create a diverse range of believable interactions for any number of individual agents. The main challenge with generating automated responses and dialogues is to make the dialogue contextually relevant and believable. Unrealistic character behavior can also break the player immersion. The creation of an appropriate solution to all these issues is an active field of research in general Artificial Intelligence, this project will limit itself on the task of creating a system for emotional transformation of utterances. An utterance in this case refers to a
single uninterrupted sequence of words and sentences. The utterance ends when its the other participants turn to say something (e.g. agent to user and the other way around). Utterances can vary from a single word to multiple sentences.

The complete automation of dialogues would open up a new question in video games. Horswill (2014) for one argues that we would need new game mechanics and genres for this kind of game systems to work.

But the need for emotionally expressive agents is not limited to video games. There is an interesting area emerging with the current rise of virtual assistants. Systems like Amazon’s Alexa or Google Home have found their way into our everyday lives. These systems currently mostly respond with a neutral sounding utterances. These systems could be made more human like with an internal emotional state or mood combined with the ability to express these emotions in their utterances. Then there would be the question of, that being something we want. This question will not be further explored in the context of this project, but it is something worth considering.

1.2 Objectives

The goal of this project is to create a system which can transform sentences from a neutral sounding form to any other pre trained emotionally toned form. This approach is also referred to as style transfer, as described by Kabbara and Cheung [2016] and Prabhumoye et al. [2018]. This means, that a system is trained to map the style or in this case sentiment of a dataset (i.e. corpora) onto a given input. The output of this model would then have the meaning of the input, but the style or sentiment of the dataset.

This project created a parallel corpora and train different recurrent neural network (RNN) models on it. The best performing neural networks then transfered the sentiment of a given set of different emotions onto the dialogue options of a virtual agent in a game setting.

The agent and the virtual environment (i.e. the game) has been created in a previous project. The non player character (i.e. the agent) shortened to NPC, offers a dialogue
tree based interaction model. A set of dialogue boxes can be used as inputs and the output is a synthesized spoken voice output.

The evaluation involves the neutral dialogue outputs and a set of emotional utterances for the agent to say. The participants got to test all the variances and have afterwards been asked a number of questions in the form of a questionnaire. The questionnaire used a Likert scale from 1 to 5 to inquire the perceived tendencies. The questions ranged from the participant’s enjoyment to the perceived emotional mood of the agent.

Two main questions were then explored with the use of the gathered evaluation data. The first question was if the users could correctly identify the intended emotion of the agent and the second question was if they enjoyed the given emotional agent more than the neutral one. These two questions were asked for each non-neutral emotion.

The results have been used to evaluate the feasibility of this NLG process compared to the traditional hand-crafted approach.
Chapter 2

Literature Review

This chapter will talk about some available technologies for use in this project. There will also be a section describing similar projects, how they were implemented and how they differ from the implemented system.

2.1 Emotional Models

Emotional models are used to understand and emulate human emotions. The most famous of which is the OCC model.

2.1.1 OCC

The OCC [Ortony et al., 1990] model, named after its authors Ortony, Clore and Collins, is the standard model for emotional synthesis. This model distinguishes 22 emotional categories as seen in Figure 2.1. Each emotional response process goes through five phases to produce an appropriate response as described by Bartneck [2002]. These phases are:

- Classification
- Quantification
- Interaction
2.1.1.1 Classification

In the classification phase the agent evaluates a given event or action and determines which emotional categories are effected by it. An example of phase can be something as simple as the user giving the agent an item. The agent needs information to properly process this action.

First it needs to know its relationship to the user (likes or dislikes) next it needs to know what this action means for the user, which is defined in the user model. The user model defines what the agent knows about the user and can therefore determine if the action was one of selflessness (if the user likes the given item) or one of indifference (user does not care or like the given item).

The next knowledge piece it needs is its own goals. The goals can be something like staying alive or accumulating something like wealth. Staying alive can be fulfilled when the given item is water and wealth can be increased if the item is money.

The next part of the evaluation process is the standards component of the agent. This basically defines what the agent perceives as normal. This means the lower the expectancy

![Figure 2.1: The original OCC model [Bartneck, 2002].](image)
of receiving something or something happening the greater the emotional reaction. An example is, if the user rarely gives away money the more relevant the action becomes. This can also go into a negative emotional reaction, if for example the agent is not used to receiving insults the more devastating they become.

The last information is its attitude towards the action or item itself. This also influence the outcome of the emotional categorization of the action.

Now the question is how to store these pieces of information. Bartneck [2002] suggests the use of an exhaustive table for limited world applications. This would be sufficient for a conventional game setting as user interactions tend to be limited. But our application will be using an open dialogue system with any number of different user utterances which the agent will need to react to. Another approach Bartneck [2002] suggests is to use abstraction. This would require us to put certain related objects/actions to be put into one category. An example for this could be to put all forms of aggressive behavior into one overarching category, differing only in intensity.

There are some [Bartneck, 2002] who believe that 22 emotional categories are more than would be needed in an agent arguing that that the OCC module was build as a representation of the human emotional spectrum and agents with limited interaction and expression capabilities might profit from a reduction of emotional categories. This might be the case for this project as the agent in this system will only be able to present a small number of emotional expression close to a single digit number range.

2.1.1.2 Quantification

The quantification phase defines how impactful an action/event is. This is done with the use of a hierarchy structure defining the goal values. The higher an action is in the hierarchy the stronger its impact on the agent. This could be for example giving the agent two items. The agent will be happier about the one item it likes more than the other one it feel less enthusiastic about.

There is another element that can play a role in the quantification and that is the history function. The history function helps to balance the emotional statue towards repetition. The agent should not react to the same an action in close repetition the same way every time. If the agent repeatedly receives the same item, it should decrease
the emotional impact of this action every time. The original OCC model does not have a history function, but the history function can be used to calculate something else that the OCC model need. The OCC model has a likelihood modifier for its emotional state representation. The history function can be used to calculate the likelihood. The likelihood is calculated by looking into its history and calculate how likely it is that the same event happens again. The use of a history function can be seen in Egges et al. [2003].

2.1.1.3 Interaction

The interaction phase serves a point of interaction with the current emotional state. This means that the impact value of actions/events need to be adjusted to fit the current emotional state. An example for this is, the fulfillment of a goal (e.g. giving an item) should not make the agent happy, when the agent is currently angry, it should just improve its emotional state.

2.1.1.4 Mapping

The mapping phase involves the mapping of the emotional categories to expressions. There might be more emotional categories than an agent has capabilities of expressing them. If the agent uses a face to express the emotions it might end up being limited to six expressions of which one, surprise, is not even an OCC emotion, reducing the expressiveness to five. To further reduce this, all 11 of the positive OCC emotions can be reduced to one facial expression, the smile Bartneck [2002]. These factors have be considered when building an emotional model.

2.1.1.5 Expression

There are multiple things to consider when expressing emotions in agents. One of which is the transition of expressions. Expression transitions need to fit what we would expect from humans. Hard cuts tend to be rare and emotional changes have to gradually transition from one expression to another.
Something else to consider is that the agent needs to show an appropriate level of expression. Not every action should lead to an extreme reaction.

Another thing is the consistency of the agent. The agent should not be cooperative with an angry look on its face, and a scared voice. The expressions need to fit one another.

2.1.1.6 Conclusion

The OCC model proves do be a comprehensive model for synthesizing human emotions. But it could be necessary depending on the application to reduce the number of emotional categories to a more manageable number. Not all of the 22 categories can be expressed by most agents.

The OCC model is despite its capabilities not enough to synthesis believable and interesting characters. Therefore you need something to create agents with character. That’s where personality models come in. Personality models allow you to create agents with certain personality traits, making each agent distinct from one another.

2.2 Personality Models

Personality models are used to emulate personality traits. There are personality models like the PEN model [Eysenck, 1990] or the better known OCEAN model [Saucier and Goldberg, 1996]. Personality models can be broken down to a number of personality traits (also sometimes referred to as dimensions) ranging between two numbers. In the case of OCEAN these would be the traits Openness, Conscientiousness, Extraversion, Agreeableness and Neuroticism. Each of these traits range from 0 symbolizing the absence of the trait to 1 the maximum presence of it [Egges et al., 2003].

The personality model can be used in combination with an emotional model to create more believable characters in an simulated environment. Egges et al. [2003] has shown how such an application could work by combining the OCEAN model with the OCC model they created two agents with the same OCC model but highly contradicting OCEAN models. The two agents behaved in a completely opposite ways to the user interaction.
2.3 Dialogue management

Dialogue systems have multiple components, each of these components do a specific task. The dialogue manager (DM) is the component that processes what has been said and contextually decides what to say. It processes the incoming information and sends a request to the next component in the line. The DM gets its information from a natural language understanding (NLU) module, which in turn gets its input from an automatic speech recognition (ASR), as visualized in Figure 2.2.

The natural language generation (NLG) element receives the contextual information from the dialogue manager and is tasked with the generation of a response. There are different implementations of dialogue systems.

![Figure 2.2: A basic dialogue interaction architecture](Aria-agent.eu, 2015)

2.3.1 Rule based dialogues

There is a multitude of rule based dialogue systems. There are finite state machines with tree structure for one. Another one is AIML [Galvao et al., 2004].

AIML is a XML based system that allows does both DM and NLG at once. It does it by having specific predefined responses to predefined requests. This means if the user says $A$ then say $B$. There is no additional step required. Its use can be extended to some degree with variables, but it does grow in complexity as each previous conversational topic has to remembered to generate prolonged coherent interactions. This makes AIML cumbersome to work with as it requires each possible request to be remembered, in some way, for it to provide satisfying responses.

This approach tends to be quite labor intensive, as each emotional state needs its own handcrafted response. There are certain ways a fairly simple system can give the impression of being more elaborate than it actually is like the ELIZA system [Weizenbaum,
that heavily relied on the use of questions and making the user do the talking. The ELIZA approach is not always feasible, especially if the user tries to get information from the agent or get the agent to do something.

### 2.3.2 Dialogue trees

Dialogue trees are another form of dialogue creation. This approach differs from the rule based one in that it provides a consideration to previously said utterance in that each state of the conversation is represented by a node within the dialogue tree. Dialogue trees are more transparent and manageable for prolonged conversations where previous inputs are relevant later in a conversation.

The downside of dialogue trees is that while they are easy to manage for smaller conversations, they tend to become larger and therefore more difficult to manage the more steps or notes the dialogue has [Rich and Sidner, 2012]. This is a common problem in any form of tree based model.

### 2.4 Style transfer

#### 2.4.1 Sequence-to-Sequence systems

A common approach to style transfer is the use of sequence-to-sequence (seq2seq) models, as described in Li et al. [2018], Niu and Bansal [2018] and Prabhumoye et al. [2018].

Seq2seq models work by encoding and decoding sequences. The encoder encodes a sequence of inputs and the decoder decodes a sequence of outputs, therefore the name *sequence-to-sequence*. In a translation task you would have a labeled dataset with each sentence in language A having a counterpart in language B. The encoder uses the language A input for training and the decoder would train on the language B input. The encoder and decoder are both connected by the last node or sequence unit of the encoder and the first node of the decoder. The encoder transfers its state into the decoder while training and the encoders output is discarded, while the output of the decoder is used as the final result (i.e. target sentence), as visualized in Figure 2.3. It is also common practice to invert the input sequence, as the close proximity of the first words of input A to the first words of input B further improve the results [Sutskever et al., 2014].
Figure 2.3: Sequence-to-Sequence model [Sutskever et al., 2014].

Seq2seq was popularized by the success of Sutskever et al. [2014]. The project showed that seq2seq models could outperform other models, when it came to translation tasks. The task of translating a piece of text to another language can be considered as fairly similar to the task of transforming one text into a slightly different text by changing the style of the text as Jhamtani et al. [2017] has shown with its modern language to Shakespearian language model. Recently seq2seq models are being more and more used in style transfer tasks.

Seq2seq models commonly use recurrent neural networks (RNN) as input and output layers. But the standard RNN suffers from a vanishing gradient problem [Hochreiter, 1998]. This makes it difficult to train models with RNNs for longer sequences of inputs as the gradient and therefore the improvement of the model decreases with each additional node. Here is where long short term memory (LSTM) networks and gated recurrent units (GRU) come in. Those have the capability of training longer chains of sequences while still being able to improve on each node with the use of an internal memory state.

2.5 Dataset

There are a multitude of different datasets, also known as corpora. There are for one unlabeled and labeled ones. Unlabeled corpora are datasets without a target value. Each sentence in a book can be regarded as a corpora of unlabeled data. Labeled data or corpora are datasets with a target value (i.e. its label), the target value can be any form of data it can be a numeric value or be character based.

This project would profit from a specific type of labeled corpora, a parallel corpus. Parallel corpora are datasets where the label has the same meaning as the other values in the data line, but in a different way. A language dictionary is a good example of one
such parallel corpus. A word in dictionary can be regarded as an input or request to a label, with the label being the same word in another language.

2.5.1 Sentiment dataset

The ideal dataset for this project would be a parallel corpus, as mentioned before. But there is no such dataset with a sentence in one sentiment, like angry and the same semantic sentence in happy or sad. There are multiple sentiment datasets with labels. The labels in these datasets are numeric values for scoring the politeness of the request data.

The Yelp dataset is used in multiple related projects [Prabhumoye et al., 2018], [Li et al., 2018] and [Shen et al., 2017]. This dataset consists of restaurant reviews where users have scored each review with a up or down vote. Up is for positive and down for negative, the review is considered to have a positive or negative sentiment depending on the number of votes. Li et al. [2018] created a similar dataset with Amazon reviews.

Another similar dataset is the Stack Overflow Politeness Dataset created by Niu and Bansal [2018] for their politeness style transfer project. This dataset consists of questions from the stack overflow platform for technical questions. The questions have been labeled by five users with a numeric score for politeness. Niu and Bansal [2018] created also a dataset with Wikipedia admin requests.

2.5.2 Creating datasets

There is also the option of creating our own dataset if there is no appropriate (parallel) dataset. One could write the data oneself, which is time intensive or crowd source the data gathering.

Crowd sourcing can be done in one of two ways. The more efficient way could be by uploading a set of unlabeled data onto a commercial platform specialized in data labeling like Amazon Mechanical Turk (MTurk) like done by Niu and Bansal [2018].

Another cost-free option could be to upload it onto a free platform and find people who are willing to label the data. A platform like Google Forms would suffice for this task.
2.6 Speech synthesis

There is a large number of synthesized speech providers (i.e. TTS systems). Most of them support an XML standard called Speech Synthesis Markup Language (SSML) for expressing emotions.

IBM Watson’s TTS [2018] is one such system. It provides a wide emotional range of expressions and uses SSML. IBM Watson’s TTS is frequently used by other projects like the one in Sorin et al. [2017] or Song et al. [2018].

2.7 Related Work

2.7.1 Style transfer

The field of style transfer is an active research field with frequent publications. There has been a rising research interest, ever since the sequence-to-sequence (seq2seq) model of Sutskever et al. [2014] outperformed all other translation tasked systems. After that the idea of using a seq2seq model for transferring styles wasn’t too far off. There have been projects which used a seq2seq model to change a normal English text into a Shakespearean style text [Jhamtani et al., 2017] and there have been many more.

2.7.1.1 Delete, Retrieve, Generate

Li et al. [2018] created a style transfer model for transforming a positive restaurant or product review into a negative one and vice versa. Li et al. trained their model on movie captions, Amazon and Yelp reviews. They used a verity of different models involving LSTM based seq2seq models. One of the things they did was calculating the appearance of each n-gram phrase in each sentiment to determine the relevance of each phrase and word for a given sentiment (positive or negative).

The first model is called the retrieve only model and is tasked with retrieving a similar sentence from the opposite sentiment. For this the system looks for the appearance of certain words in the other sentiment dataset.
The second model is called the \textit{template-based} model. This model deletes a phrase which it associates with its current sentiment and retrieving a replacement phrase from the other sentiment. The system retrieved phrases which appeared in a similar context. This sometimes results in grammatically incoherent sentences as the model tries to insert the replacement into the same place where it deleted the previous phrase.

The third model is called the \textit{delete only} model and is a trained seq2seq model. It uses both the initial sentence and the retrieved phrase from the second model to create a syntactically correct sentences.

The fourth model is called the \textit{delete and retrieve} model and is also a trained seq2seq model. This model differentiates from the third model in that it uses the retrieved replacement sentence from model one instead of the replacement phrase from model two as input, with the initial sentence to produce a completely new sentence instead of replacing a phrase.

A 1 to 5 Likert scale human evaluation showed that all their models performed better on average than previous models for the given dataset. The evaluation was split into the three datasets and three criteria, grammatical correctness, content preservation and attribute match (positive/negative) for each dataset. The fourth model clearly outperformed the other models, whilst the other models performed fairly similar to one another.

\subsection*{2.7.1.2 Polite Dialogue Generation Without Parallel Data}

\citet{niu2018polite} attempted to generate polite and rude responses to requests. This project approaches the challenge of style transfer differently, as its goal is not to transform, but generate new unrecorded responses. This project also works without parallel data.

The polite dialogue generation project uses Wikipedia admin requests, stack overflow requests and the MovieTriples dialogue corpus \cite{niu2018polite} as training datasets. The project used a variety of different models, one was a simple seq2seq model \cite{bahdanau2014neural}, another one was a fusion model which was an adapted seq2seq model involving a language model which was exclusively trained on polite utterances. The
third model was a label-fine-tuning (LFT) model which allows the model to produce a more varied result as it has an additional input value for adjusting the level of politeness.

The last model they tested was a reinforced learning (RL) model. The RL model used an additional model for classifying its results and adjusting the loss value for each training set. A LSTM and CNN (constitutional neural network) model was used to classify the produced utterances into polite and rude.

The evaluation was done by anonymous participants on the MTurk platform, the results show no significant improvement of the before mentioned models over previous models, the RL and LFT models did perform slightly better in some categories than the other two models.

2.7.1.3 Style Transfer Through Back-Translation

Another project involving style transfer was done by Prabhumoye et al. [2018]. This project applied style transfer in three different ways, sentiment, gender and political slant.

The project used a gender annotated Yelp review dataset for sentiment and gender style transfer. The political slant dataset came from Facebook, only top-level comments directed to United States senators with their political affiliations where considered.

The models of the project consist of a mix of LSTM, CNN layers and adversarial components.

The overall result showed a higher level of fluency than the baseline projects, but the authors do acknowledge that the meaning of the utterances where not always preserved.

2.7.2 Emotionally expressive agents

Over the years there have been other projects which attempted to use a more varied approach to emotionally adaptive agents. Some of them will be discuses in detail. Façade is a game, while ERiSA, ARIA VALUSPA and FAtiMA are frameworks that can be implemented to other systems.
2.7.2.1 Façade: Handcrafted Emotional Output

Façade has been made by [Mateas and Stern, 2003] and is a drama game in which the user controls a character who is entangled in a domestic quarrel (see Figure 2.4). The main interaction with the NPCs happens over typed text input, while the NPC output is a playback of prerecorded dialogues by voice actors.

![Figure 2.4: Façade interactive drama dialogue example [Mateas and Stern, 2003]](image)

Façade uses a rule based system for its dialogue. Conversational topics are split into what Façade refers to as beats. There are two types of beats, local and global ones. Local beats in their number of interactions but can be as short as single transaction like asking the player for her/his view on a isolated subject. While global beats consist of multiple local beats, with a certain story goal. An example of a global beat is, when one of the two characters in Façade is trying to make the other one look bad by telling the player multiple (local beats) stories of how neglecting, absent or rude the other character is, all the while asking the player for approval. Beats have preconditions, weights, priorities and effects.

Each player input is categorized into one of 36 predefined expressions. This limitation can sometimes lead to false positives, where the player seems to approve of something even tough she/he didn’t.

Façade uses something called a drama manager to create engaging stories. The drama manager is a dynamic system that tries to create an engaging custom experience for the user. The drama manager recalculates the story after each story beat. Beats are self contained sub-stories, which influence the overarching story. The drama manager uses
these beats to create an optimal tension arc for the overall story. The drama manager chooses beats by a number of rules, the fulfillment of their preconditions, their weights and priority, which defining their overall likelihood. Each finished beat also effects other beats likelihood of being selected, this depends on the outcome of the beat, the outcome is effected by something like the players approval or disapproval.

The creators of Façade spent over two years just on writing the dialogues. All of the interactions in Façade are hand crafted, from the dialogues to the facial expressions and voice recordings. There is always a limit in the range of conversations you can cover with hand written dialogues. Currently the hand crafted approach delivers an overall better player experience compared to automated alternatives, but this comes at the price of human labor.

2.7.2.2 ERiSA: Using the OCEAN model

ERiSA is a framework for emotional agents, that can detect and interpret users verbally and non-verbally [Chowanda et al., 2014]. The framework facial (i.e. non-verbal) processing allows the system to remember users and recognize their emotions.

The ERiSA framework consists of the following five components:

- The **sensing component** for STT and facial recognition
• The **interpreter component** for processing the provided data from the sensing component.

• The **agent component**, which contains the internal state, interaction rules, personality of the agent and the social relationship towards the user.

• The **game component**, containing the game state and game rules.

• The **behaviour component**, which contains the action manager and action selector. The behaviour component gets its input from the interpreter (user behaviour), agent (attitude toward the user) and game (game rules) component and decides a course of action. The behaviour component updates all the states in the relevant components after its action.

The emotional state of the agent is reflected in its social relationship value, which is calculated in a number of steps. ERiSA uses the OCEAN personality model [Saucier and Goldberg, 1996] in combination with an average event value to first calculate the emotion value, which is then used in combination with a decay function to get the *like* value. Then the *like* value is multiplied with the *knows* value to get the social relationship (*R* value in Figure 2.6).

![Figure 2.6: Game flow. Action streams are selected based on the social relation value *R* which is either 'very close', 'close'. 'Intro' is used when the user is unknown [Chowanda et al., 2014].](image)

The ERiSA framework uses a handcrafted rulebased dialogue system for its user interactions. ERiSA has been used in a number of game settings. One of which was a simple don’t smile video game [Chowanda et al., 2014], where the challenge was getting the other player (in this case the user) to smile or laugh, with the use of jokes and funny facial expressions. The dialogue flow is visualized in Figure 2.6, a non game setting would look the same, only without the game related steps.
Another game setting was a custom-built quest in *The Elder Scrolls V: Skyrim* [Chowanda et al., 2016]. The *Skyrim* implementation used a conventional dialogue box selection as user inputs, instead of a speech driven open dialogue system.

ERiSA uses a rule-based dialogue system where each interaction has to be pre-scripted. This form of dialogue generation is time intensive and static.

### 2.7.2.3 ARIA VALUSPA: Extending ERiSA

ARIA VALUSPA [Aria-agent.eu, 2018] is a framework consisting of an information retrieval agent that can process natural language, recognize social cues and express personality traits. ARIA has a modular architecture with three main blocks [Valstar et al., 2016] as seen in Figure 2.7. The first block is the input block which processes the recorded audio and video. The second block is the agent core tasked as the name suggests with handling the all the information and states to generate appropriate responses. It receives its informations from the input block and hands its responses to the output block which generates the agent behavior for the user to perceive.

![ARIA framework architecture](image)

**Figure 2.7:** ARIA framework architecture, composed of modular Input, Agent Core, and Output blocks [Valstar et al., 2016].

ARIA is build with multiple preexisting frameworks. It uses ERiSA as a base, Living Actor for the facial animations and Cereproc for its voice synthesis [Aria-agent.eu, 2015]. The dialogue manager is implemented as a probabilistic graphical models which uses a Dynamic Bayesian Network to determine the next dialogue options.
2.7.2.4 FAtiMA

FAtiMA is a modular framework for emotional agents. Its planning capabilities allow it to make decisions based on its emotions and personality [Dias et al., 2014]. The main algorithm of the framework is its core. The main component of the framework is the FAtiMA core, which can be further extended with its modular components (Figure 2.8). In FAtiMA perceived actions/events are stored in the memory and go through an appraisal process. The result of the appraisal is used in combination with the history of previous actions (from memory) to produce an action.

![Figure 2.8: FAtiMA Core Architecture [Dias et al., 2014].](image)

The FAtiMA core uses a two step appraisal system that is strongly leaned on the work of Marsella et al. [2010]. The first step of the appraisal process evaluates the relevance of an event for the agent and sets appraisal variables, while the second process generates an affective state from those variables.

The core can not do anything on its own and need modules to work. One such module is the reactive component which matches events to a set of emotional rules. Another one is the cultural component, which changes the values generated through appraisal depending on cultural affiliation. An example of a cultural trait is that agents who are collectivistic value events involving selflessness more praiseworthy than agents without that trait. If for example one agent gives another an item that the other needs. This results in the one giving no longer being in possession of the item and the receiver who was in need of the item to be thankful. There are multiple other modules, who value behaviors differently.
The FAiMA framework offers a sensible approach in emulating human emotions. It is a rule based system that requires a set of predefined and emotionally annotated interactions to work. The FAiMA framework doesn’t do much on its own, as mentioned before. There is a need for other applications to be added to display complex behavior, like dialogue components.

2.7.3 Evaluation of emotional output systems

The before mentioned ERiSA smile game implementation [Chowanda et al., 2014] used personality traits for their evaluation. The system was first tested against itself, by creating two agents with different personality settings. The first agent (Poppy) had a high extroversion score while the second (Spike) had a high neuroticism score. Their test showed that the agent with the high neuroticism score had an easier time defeating the other agent as seen in Figure 2.9.

![Figure 2.9: Smile Simulation. The blue and purple lines show the character’s urge to smile for Poppy and Spike, respectively. The orange line indicates the threshold to display a small smile and the purple line the threshold for smiling uncontrollably [Chowanda et al., 2014].](image)

The second test involved human participants. The participants played the smile game against both before mentioned agents (Poppy & Spike). The participants where then asked a series of questions including how they valued high they valued certain personality traits of the agents. The participants where able to pick up on the implemented differences. They valued Poppy twice as extroverted as Spike and Spike twice as neurotic as Poppy. The test proved that the system was able to create agents with distinct character traits which humans are able to recognize.
2.8 Summary

There are many technologies approaching different aspects of making agents more engaging or easier to program. There are also projects more focused on transferring styles and sentiments in a generative way.

The ARIA VALUSPA system is one such. It uses the ERiSA framework to produce emotionally toned dialogues and also uses Cereproc for emotional speech synthesis and Living Actor for its facial animations to express itself [Aria-agent.eu, 2015].

The OCC model is a comprehensive model of the human emotional landscape. The OCC model could be used to synthesize the emotional state of agents.

The OCEAN model can be used to create the personalities for agents. A emotional model like OCC is not enough on its own to produce a distinct emotional agent as a personality does not just consist of a emotional component. A believable character needs both an emotional model like OCC and a personality model like OCEAN. OCEAN was successfully used in the ERiSA system [Chowanda et al., 2014] to produce distinct characters, but the ERiSA used a simpler relationship function as its emotional model.

The systems discussed in the emotionally expressive agent subsection in this chapter are all using rule based dialogue systems, which consist of handcrafted dialogues. Handcrafted dialogues are a time consuming way of producing dialogues. Each emotional state requiring its own strings. They are also context dependently, this means they written with a specific application in mind and are hard or at least not instantly transferable to other settings or applications and need to be adjusted.

Sequence-to-Sequence systems provide an alternative to the handcrafted systems as they can generate any number of utterances from a given utterance and the intended sentiment. This kind of system can be trained on any corpora.

The evaluation process used in Chowanda et al. [2014] provides a reasonable way of testing the systems emotional components. Some variance could be added to this evaluation by increasing the number of agents involved or by adding additional components like the facial animations, emotional voices. The users can also be asked to value the quality of each of these components or/and the quality of the produces dialogues.
Chapter 3

Methodology

3.1 Data collection

The data collection for the training of the models was done over online crowd sourcing. Participants could voluntarily contribute to the data via a the Google Forms [Google, 2018] platform. Each participant was asked to transform 20 neutral utterances into three emotional variances. The target emotions where angry, happy and sad. A total of 8 participants contributed.

3.2 Emotional NLG

The emotional NLG process needed to create a range of emotionally distinct strings. This process explored a variation of Sequence-to-Sequence models [Sutskever et al., 2014]. A large amount of time was spent researching different NLG processes. A number of different RNN models were considered for this project and the final implementation used both LSTM and GRU recurrent units (see Chapter 4).

The RNN based Sequence-to-Sequence models were implemented using Python with the Keras [2018] library and TensorFlow [2018] as the back-end. Both libraries are commonly used for neural network applications.

The created models where then trained with the collected data from the online Form.
3.3 Model integration into testing environment

The testing environment for this project was a game that was developed in a previous project [Agnus et al., 2018]. The game included the IBM-Watson TTS [2018] system for speech synthesis, a graphical user interface with a character model for the agent and a tree-based dialogue system.

Some adjustments had to be made to the game for it to work with the trained models. Some of the changes involved the integration of the generated emotional utterances into the system (more details in Chapter 4).

3.4 Experiment

The participants were informed of their rights and signed a consent form. Each participant played the three integrated game modes in random order, those being neutral, angry and happy. After each mode the participants filled out a Lickert scale questionnaire.

3.5 Results

The results compared the two emotional modes to the neutral one. The average score was used to identify the tendencies and the McNemar test was then used to calculate the statistical significance for these tendencies.

The results show that users could distinctly identify the angry agent while happy was more challenging. There was no statistically significant evidence that users enjoyed the emotional modes over the neutral one.
Chapter 4

Implementation

The implemented system went through three main stages in its development.

- the creation of the dataset
- the development of the RNN model
- the integration of the neural net into the game system

4.1 Dataset

The initial goal for acquiring a dataset was to find a preexisting sentiment related parallel dataset. The decision for using parallel data over non parallel labeled data was taken because of the fact that models trained on parallel data clearly outperform models trained on labeled nonparallel data. This can be seen when comparing the results of Sutskever et al. [2014] for language translation tasks and Jhamtani et al. [2017] for Shakespearean writing style transfer, to Li et al. [2018] and Prabhumoye et al. [2018] works in sentiment transfer.

The conclusion was after thorough research that there was no usable parallel sentiment dataset. Most projects have used scored labeled datasets with reviews or requests [Li et al., 2018], [Niu and Bansal, 2018] and [Prabhumoye et al., 2018].

The consequence of not finding an appropriate dataset was the creation of such. The decision was made to go for a free crowd sourcing approach over a payed solution as
there was no dedicated budged allocated to the project. There are multiple possible approaches to this, but the most straight forward one was to create a online form and request volunteers to participate in it.

4.1.1 Data form

Google forms [Google, 2018] was used as the sole data gathering tool for this project. Google forms is a free platform for forms. The form process was completely anonymized and contained no personal data of the volunteers.

The form consisted of twenty entries (see Appendix A). Each entry contained a written utterance in a neutral sounding manner and users where expected to write down angry, happy and sad sounding versions of the given utterance. There where no mandatory fields and users where encouraged to not force themselves to find an appropriate counterpart for the given emotion. The two emotions with the highest entries (happy and angry) where then decided to be trained upon for the implementation.

The utterances in the form are domain specific sentences. Domain specific means that the utterances are somewhat similar, but not identical, to the final test set on which the game system has been trained on. This is necessary as the number of possible entries in our case twenty, is to low to allow a more general approach.

4.1.2 Gathered data

Eight users volunteered to fill out the form. The resulting data is by its very nature unbalanced. As there are only twenty entries in the neutral column appearing multiple times with different labels in the emotion columns. But there seemed be no other time saving alternative as the data gathering process was not to be the main focus of this project.

The gathered data had some duplicates in the emotional columns too, these where deleted to reduce the risk of one utterance being to dominant. The emotional label sad had the least entries, even before the data cleaning, as most people seamed to have difficulties finding sad ways to express neutral utterances.
After the cleaning process the dataset consisted of 128 angry and happy entries and 118 sad entries. The happy and angry emotions were taken for the training of the model as they had the highest number of entries. The dataset was then extended by a couple more lines bringing the total of angry to 169 and happy to 164.

### 4.2 Sequence-to-Sequence model

Four different models were trained in this project. One for each embedding and one for each recurrent unit. The first embedding is a character based one and the second one is a word based. The recurrent neural net (RNN) units are for one the long-short-term-memory (LSTM) unit and the other is the gated recurrent unit (GRU). The aim was to find the best performing models to use.

All the models have been created in Keras [Keras, 2018] with TensorFlow [TensorFlow, 2018] as the back-end. Keras provided a sufficient programming interface for the project’s RNN-based model.

#### 4.2.1 Design

The general design of the implemented seq2seq model consists of five layers. It works like the one visualized in Figure 2.3. The first and second layers are the input layer for both the encoder and decoder, which run in parallel. The third layer is the encoder layer which consists either of LSTM or GRU units. This layer sends the state of the sequence chain to the next layer, the decoder layer also consists of LSTM or GRU units (depending on which model is used). The decoder then uses both its input layer and
the encoder’s state value to produce an output sequence. The last layer is the output layer which returns the a sequence. The process is visualized in Figure 4.1, which is the GRU based character embedded model. The other models only vary in the size of their inputs and outputs and not in the overall design of the layers.

The code (build on the model of Chollet [2018]) for the GRU layering:

```python
encoder_inputs = Input(shape=(None, num_encoder_tokens), name="Encoder_Input")
encoder = GRU(latent_dim, return_state=True, name="Encoder_GRU")
_, encoder_state = encoder(encoder_inputs)

decoder_inputs = Input(shape=(None, num_decoder_tokens), name="Decoder_Input")
decoder_gru = GRU(latent_dim, return_sequences=True, return_state=True, name="Decoder_GRU")

decoder_outputs, _ = decoder_gru(decoder_inputs, initial_state=encoder_state)

decoder_dense = Dense(num_decoder_tokens, activation='softmax', name="DecoderOutput")
decoder_outputs = decoder_dense(decoder_outputs)

model = Model([encoder_inputs, decoder_inputs], decoder_outputs)

model.compile(optimizer='rmsprop', loss='categorical_crossentropy', metrics=['accuracy'])
```

The 45 value in the InputLayer seen in Figure 4.1 stands for the vocabulary vector, it is in this case the number of different characters, but could also stand for the number of different words within the given label of the dataset. The 45 is preceded by two ”None” parameters. None stands for variable size. The first of the two None parameters is a stand in for the size of the full batch. There is no need to further split the batch into mini batches as the dataset is very small. The Second None parameter is for the length of the utterance and varies depending on which embedding is used. The word embedding consists of less utterance sub elements (words) than a character embeddings sub elements (characters). A utterance with 90 character might only consist of 20 words.

### 4.2.2 Embedding

This project uses two different types of embeddings, the fist one is character embedding and the second one is word embedding.

The embedding can be further specified in how words and characters are embedded. One-hot embedding is the one being used in this project. The one-hot embedding approach consists of a large vector consisting of all the possible input values and sets
the actual input value within the vector to one and everything else to zero. All possible input values would be in the context of a character embedding all possible characters, including white spaces, numbers and punctuation marks. Each character is represented by one value in the vector. If we take the example of the word "my" then there would be two vectors with each vector having a single one entrance at different positions, as "m" and "y" are different characters. The character embedding vectors within the project consist of up to 53 entries for each vector.

The word embedding works in a similar way with the only difference that each entry in the vector represents a word and not a character. This makes the embedded vector much larger but the number of input nodes much smaller, as there are viewer words than characters in a sentence. The largest number of words or one could say vocabulary, is in the output layer with a vector size of 263. The 263 stands for the number of different words or entries within the vector.

There was also the option of using the word2vec approach like the one used by Niu and Bansal [2018]. The issue with the word2vec approach is that it is not applicable to character embeddings, as characters don’t have the same relationship to one another as words do. word2vec also requires extensive training or at least the use of a pre-trained vector model. The small is not ideal for this approach.

Both the encoder and decoder have different vector sizes, as they where both trained with their respective vocabularies. In the case of the encoder, only the neutral data was used to create the encoder word/character vector and the same goes for the decoder (angry and happy data). This greatly reduced the size of the word/character vector, without having a negative impact on the overall model as a general vector which included all vocabulary would have had dead units/neurons as those would never be triggered in the given context (e.g. curse words in the input).

4.2.3 Recurrent unit

The nature of the project required the use of an RNN based model. Multiple different RNN unit types where implemented in an attempt to find the best performing model.
4.2.3.1 Long-Short-Term-Memory

The first type of RNN units used were LSTMs. LSTMs have two internal memory states, the hidden and cell state, these are transferred to the next sequential unit within their own layer and one output value that is transferred to the next layer. The cell state is only slightly adjusted each unit and serves as the layers long term memory, while the hidden state serves as the units short term memory. LSTMs are seen as the standard for translation and style transfer tasks, as done by Sutskever et al. [2014], Li et al. [2018] and Kabbara and Cheung [2016], just to name a view.

4.2.3.2 Gated-Recurrent-Unit

GRU is a fairly new RNN unit type [Cho et al., 2014] and therefore has not been used as frequently as LSTM. But some [Chung et al., 2014] experiments have shown that it can perform similarly good and in some cases better than LSTM, despite being computationally less expansive.

GRU differentiates itself from LSTM by only having one internal memory state compared to LSTM's two states. GRU therefore also has a less complex internal workings and a reduced number of function calls (as seen in Figure 4.2).

4.2.4 Training results

Each model was tested in a variety of different ways.
4.2.4.1 Training speed

Each of the models was tested with both a CPU (quad core Intel i7-6700HQ) and a GPU (NVIDIA GTX 965M). All the models for this experiment have been trained on the angry dataset as initial tests have shown that the angry and happy datasets returned insignificantly different scores, this probably stems from the similar data size (169 for angry and 164 for happy). Table 4.1 shows the average processing speed of one epoch in milliseconds. An epoch is the one time processing of all the training data. The table clearly shows that the processing speed of the word encoded models are multiple times faster than the character encoded models. This is to be expected as the character embedded models are larger overall. The input layer for example is in the case of the angry dataset 169x90x45 (entries x longest utterance x number of characters) compared to the word embeddings 196x22x121 (entries x longest sequence of words x dictionary size), the character embedding is about a third larger in all the input and output layers. This most likely leads to the tripling of the processing time. One can say that the GPU clearly performed better as it was to be expected. The character embedded model performed especially badly on the CPU, it took 500 times longer.

<table>
<thead>
<tr>
<th></th>
<th>GPU</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>word</td>
<td>character</td>
</tr>
<tr>
<td>LSTM</td>
<td>0.6749</td>
<td>2</td>
</tr>
<tr>
<td>GRU</td>
<td>0.5268</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 4.1: Processing speed of one epoch

The content of Table 4.1 clearly shows that the word embedded models perform much faster than the character models, with the GRU model outperforming the LSTM by about 22%, this would prove the findings of Chung et al. [2014], if there wasn’t the issue of the character embedded models performing equally well for both the LSTM and GRU, where there was no notable difference. But this might be do to an internal rounding function of the time in the Keras library.

4.2.4.2 Training and validation

The word and character embedded models have been tested with both the LSTM and GRU units. The results have been recorded and compared by accuracy, loss and a user evaluation, to determine which models to use for the game system integration.
The accuracy is a common equation for determining the efficiency of statistical models. It is calculated by dividing the accurately predicted results by the total number of entries in the predicted set.

The loss value represents the deviation of the predicted results from the optimal condition. Zero would mean that there where no deviation (i.e. miss matches). The loss is also the value thats being used to backpropagate to improve the model.

The training of the models used 90% of the datasets. While 10% where reserved to validate the model with utterances it was not specifically trained on.

The word embedded models eventually all reach a training loss of below 0.1 at around 500 epochs (see Figure 4.3). But the training accuracy is not significantly improved from epoch 300 onward, where it reaches a plateau with slight fluctuation. The best performing models here are the GRU models which reach an accuracy of up to 0.4 but averaging around 0.37.

The validation set on the other hand reaches its lowest loss at around 200 epochs and further increases with time (i.e. number of epochs). This most likely stems from some form of overfitting of the model to the training set, but this is to be expected, as the dataset of the project is very small (total of up to 169). It is worth noting that the LSTM models have an overall lower loss than the GRU with the validation set. The validation accuracy does not significantly decrease after epoch 200 despite the increase in loss. The improvement of the validation accuracy in all models stagnates at around 400 epochs, with the GRU models having much grater fluctuations than the more gradually changing LSTMs.

The word embedded LSTM models show a more gradually learning curve (i.e. accuracy) than the GRU based models, but the GRU based models reach a higher average training accuracy of 0.35 against the LSTM’s 0.31. Especially in the validation set you can see (Figure 4.3) an accuracy jump of up to 0.13 within a single epoch for the GRU models. The final validation accuracy lies at around 0.11 for all word embedded models.

The character embedded models eventually reach a training loss of around 0.1 towards epoch 900 (see Figure 4.4). The validation loss behaves in a similar way to the one in the word embedded models with an eventual rise at around epoch 300. Both the GRU and LSTM models show a much more gradual behavior in the character embedded models.
than in the word embedded ones. The training accuracy stagnates at around 600 epochs for all the character embedded models while the GRU models do have a 0.05 better on average accuracy than the 0.30 of the LSTM models.

The accuracy improvement of the validation loss in the character embedded models stagnates at around epoch 300. The GRU models here too, deliver a slightly higher average end result with 0.16 over the LSTM’s average of 0.13.

The word and character embedded models have shown similar results for the training set with the character models being more consistent and stable in their behavior over the word embedded models. The character embedded models did outperform the word embedded models on the validation side of the evaluation, as they reached both a higher on average accuracy of 0.15 against 0.11 and a lower on average loss of 0.95 against 1.5.

The models where also evaluated by a user to determine the overall usability of the produced utterances. All the models where trained for 1000 epochs and then given the same eleven neutral sentences to transform and the user evaluated them from a score of 0 to 2.
• 0 for neither the meaning was preserved nor the intended sentiment was expressed
• 1 for either the meaning was preserved or the intended sentiment was expressed
• 2 for either the meaning was preserved and the intended sentiment was expressed

In the case of the angry dataset the character embedded clearly outperformed the word embedded systems as seen in Table 4.2. The word embedded GRU model performed especially badly with a total score of only one. This bad performance might be related to the data fluctuation of the validation accuracy in Figure 4.3 did show a more random and less knowledge based success approach.

It was decided that the best choice for the angry model in the game system would be the LSTM character model, as it reached the highest total score of 13, while never going below 1 for each utterance.

The models overall performed much better with the happy dataset, with the lowest score being 9 and the highest 18 (seen in Table 4.3). Therefore the model pick for the happy integration was the GRU character embedded model. Table 4.3 shows an example of the model outputs.
4.3 Game Implementation

The game part of the project was taken from a previous project. The game from the previous project [Agnus et al., 2018] did not have any machine learning component.

4.3.1 Initial game

The game from the previous project was developed using Unity [Unity, 2018] and has a medieval tavern setting. It had two game modes, a voice commanded version (see Figure
<table>
<thead>
<tr>
<th>Input</th>
<th>How may i help you?</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Happy</strong></td>
<td></td>
</tr>
<tr>
<td><strong>LSTM word</strong></td>
<td>got the time??????</td>
</tr>
<tr>
<td><strong>LSTM character</strong></td>
<td>Hello stranger. May I help you?</td>
</tr>
<tr>
<td><strong>GRU word</strong></td>
<td>How may i help you?</td>
</tr>
<tr>
<td><strong>GRU character</strong></td>
<td>Hey my friend. Can I do anything for you.</td>
</tr>
<tr>
<td><strong>Angry</strong></td>
<td></td>
</tr>
<tr>
<td>? had to walk all the way</td>
<td></td>
</tr>
<tr>
<td>What the fuck to you want?</td>
<td></td>
</tr>
<tr>
<td>You go. do you need more</td>
<td></td>
</tr>
<tr>
<td>I’ll get him some. This time.</td>
<td></td>
</tr>
</tbody>
</table>

**Table 4.4:** Sample one output of the models

<table>
<thead>
<tr>
<th>Input</th>
<th>She lives down the &lt;location&gt;.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Happy</strong></td>
<td></td>
</tr>
<tr>
<td><strong>LSTM word</strong></td>
<td>Is a great walk home..</td>
</tr>
<tr>
<td><strong>LSTM character</strong></td>
<td>I am terribly sorry, and you are great, but I am just not looking for a relationship right now.</td>
</tr>
<tr>
<td><strong>GRU word</strong></td>
<td>Is great. she has been</td>
</tr>
<tr>
<td><strong>GRU character</strong></td>
<td>It’s great she lives here.</td>
</tr>
<tr>
<td><strong>Angry</strong></td>
<td>Down. now!!!! see anyone</td>
</tr>
<tr>
<td>Where the fuck is it?</td>
<td>Sadly with him. i do you</td>
</tr>
<tr>
<td>Where is it?</td>
<td></td>
</tr>
</tbody>
</table>

**Table 4.5:** Sample two output of the models

4.5) where users could just say any utterance and the system would try to react to it and a dialogue tree system where the user could select a dialogue option from a limit set of utterances. The conversations happen with a bartender agent.

![Figure 4.5: The Speech-Driven Game System system architecture](image)

The game used an adapted Alana system, which was in turn build for the Amazon Alexa challenge [Papaioannou et al., 2017]. The system was able to respond to predefined user utterances. The tree dialogue system was more rigid but offered a more structured dialogue.

The system used the IBM-Watson TTS [2018] system for speech synthesizes.
4.3.2 Game adjustments

The game system was changed in a number of ways to fit the needs of this project. Only the tree based dialogue system was used for this project as the open dialogue approach with the Alana system would have proven to difficult for the users. The produced utterances from the RNN models are not always clear and the tree based dialogue system provides the option of guiding the user through the conversation by only providing them with dialogue options that progress the conversation.

Both the text generation and voice synthesis were generated offline, this means the text was processed by the models, then synthesized by IBM Watson’s TTS and both were then integrated into the game. This means that the only component left from the initial games runtime system seen in Figure 4.5 is the Game component of the Unity Engine. This was in part done to avoid long latency issues, associated with slow or unstable connections to the Internet. This issue arises because the synthesized voice files take some time to be generated and downloaded (i.e. the slower the Internet connection the longer it takes).

4.3.2.1 Game setting

The goal of the game is to find a set of ingredients to win the game. There are three game modes from which the user must first choose from, to start the game.

- Test A: Neutral agent, uses the handwritten input utterances
- Test B: Happy agent, uses the generated utterances from the character embedded GRU model
- Test C: Angry agent, uses the generated utterances from the character embedded LSTM model

The user then converses with an agent (as seen in Figure 4.6) with different emotional states, depending on the chosen Test (i.e. game mode). The game ends when the user is congratulated by the agent for finding all the required ingredients.
The dialogue in the game is kept fairly simple as the larger a tree based structure becomes the more branches it grows and the more dialogues have to be written. Therefore the decision was made to not go too deep. The maximum depth of the dialogue tree is 11 layers (seen in Figure 4.7) before it redirects to different previous node.
Figure 4.7: The games dialogue tree.
Chapter 5

Legal, Ethical and Professional issues

5.1 Legal issues

The tools used in this implementation are either under the GNU license or have been appropriately licensed from the license holder.

5.2 Ethical issues

The system is not intended for commercial use and has therefore been only used for research purposes. The implementation has been tested in an controlled lab environment and under supervision.

The participating human subjects have been required to be of legal age and made aware of any potential risk. Participants were told the risks involving, the system giving inappropriate responses, because of the generative nature of the system or any other video game related risks like motion sickness and epileptic seizures.

All participants have signed an informed consent form (see Appendix A) before the evaluation and all data gathered has been anonymized. There has been no gathering of sensitive data.
5.3 Professional issues

I uphold myself to the highest professional standards, as someone aspiring towards becoming a professional in his field.

I will keep the gathered data from my evaluations in an anonymized form and use them for only the purposes agreed on to by the participants.
Chapter 6

Evaluation and Results

6.1 Experiment design

The experiment consists of multiple steps, consisting of filling out forms and playing the game. There are two questionnaires, one "before" the first game and one "after" each game.

The participants were given a consent form, an instruction document and a participation-ID with the order in which to play the game modes. The order in which each participant played a game mode was randomized to reduce biases toward one specific mode.

All participants filled out the consent form before participating in the experiment and they were provided with additional information when they required it. Each participant did the following steps in this order, after giving their consent:

1. Filling of the "before" questionnaire
2. Playing the game
3. Filling of the "after" questionnaire
4. Playing the game
5. Filling of the "after" questionnaire
6. Playing the game
7. Filling of the "after" questionnaire
6.1.1 The ”before” questionnaire

The before questionnaire (see Appendix C) had the following questions for the participants to fill out:

- Age
- How often do you play video games? (5 point Likert scale)
- Have you played role playing games before? (yes/no)
- Have you participated in any way in an experiment involving the character in the image above? (yes/no)

The last question was used to find out if the participant was in any way involved in the previous project from which the game system itself was taken from.

6.1.2 The ”after” questionnaire

The after questionnaire (see Appendix D) mostly consisted of 5 point Likert scale questions and was inspired by the Godspeed Questionnaire series [Weiss and Bartneck, 2015]. The first list of questions was about the users latest game experience with the two opposites:

- gruesome/fun
- boring/interesting
- easy/difficult
- confusing/clear

The second round of questions was about how the user perceived the agent:

- rude/polite
- calm/angry
- sad/happy
6.1.3 Game experiment

Each participant was given a instruction (see Appendix B) and asked to play the game three time, while filling out a questionnaire after each game. The game system had the three modes neutral, happy and angry. Each participant was given a randomized order in which they attempted the game modes. The game started by displaying three buttons with the titles ”Test A”, ”Test B” and ”Test C”, each representing a game mode. Depending on the game mode the user was either confronted with a neutral, happy or angry agent. The goal in each mode was to find the location of the two ingredients, honey and butter, by asking the agent questions. The game stopped each time after congratulating the users for finding the whereabouts of the ingredients. All the games had the same dialogue options with the only difference that the agent reacted differently to the user requests. Some responses of the agent made little sense as the meaning was sometimes lost in the style transfer. All the participants where able to complete the games, because of the rigid structure of the underlying decision tree.

6.2 Results

For the evaluation of the results the mean values were calculated for each ”after” questionnaire entry, where the entries ranged from 1 to 5 on a ordinal scale. The values have been rescaled to range from -2 to +2, because the questionnaire was build with the aim to find the tendencies between two opposing words, zero is therefore a better central value than 3 for this purposes. All the values discussed in this section refer to an mean value of all the entries.

The first section of the questionnaire dealt with the question of how the users felt about the mode they just played. The visualized values in Figure 6.1 show these tendencies.
The users enjoyed the neutral mode with a mean score of around 0.85 most, with the angry mode came second with a score of 0.15 and the happy mode was enjoyed the least with a gruesome score of 0.15.

The angry mode was perceived as the most interesting one with a score of 0.61, while happy was perceived as more boring with a value of 0.15 on the boring side.

All the modes where perceived as easy with the neutral mode having the highest score of 1.85, this was to be expected as neutral was the handcrafted baseline with the most semantically accurate dialogue responses.

The same goes for the clarity score where the neutral mode also scored the highest and by calculating the correlation coefficient of a negative value of -0.686 (with the Kendall’s tau coefficient [Noether, 1981]) we can say that there is a high (negative) correlation between the perceived clearness and easiness. The angry mode was perceived as the most confusing mode with a score of 0.77, this may be due to the high tendency of curse words and a perceived unwillingness of the agent to help the user.

The second section of the questionnaire dealt with the question of how the users felt about the agent in the mode they just played. There are clear tendencies seen in Figure 6.2.
All the users clearly identified the agent in the angry mode as the rudest by giving it the maximal score of 2, while the agent in the happy and neutral where perceived as similarly rude with a score of 1.69 and 1.38 respectively. The happy agent was therefore perceived as slightly more polite as the neutral one.

The angry agent was correctly identified as angry by the majority of the users with a mean score of 1.62, while the neutral and happy agent where perceived as equally calm with a score of 1.77.

The neutral agent was surprisingly perceived as more happy than the happy agent, but neither one achieved a score of above 1 on the happiness score. Angry was perceived as slightly sad with a score of 0.69.

The neutral agent was also perceived as the most alive one, with a score of 0.54. Both the angry and happy agents received a fairly neutral score of 0.15.

The competency and intelligence scores have a high correlation with a Kendall’s tau co-efficiency of 0.689. This can be clearly seen in Figure 6.2, where the largest divergent value between these two is the angry agents mean score of 0.92 in incompetence and 0.54 in unintelligence.
6.3 Hypothesis testing

The experiments were designed and executed with the following hypotheses in mind. All the hypotheses were tested with the McNemar test as this test lets two related ordinal results be compared to one another.

For the McNemar test all the involved participants neutral mode values ranging from 1 to 5 (not the rescaled version centered around 0) where used and compared with the same participant’s feedback on the played emotion. The value 5 (for a definite identification of the given sentiment or emotion) was used as the baseline for which all entries were searched for.

6.3.1 Hypothesis I: Users correctly identified the angry agent as more angry than the neutral one.

The data displayed in Figure 6.2 already shows a high tendency towards the correct identification of this hypothesis, as angry is being strongly identified with a mean of 1.62 while the neutral agent was more on the opposite tendency, on the calm side.

A McNemar test was done to further investigate this hypothesis with the null hypothesis of there being no statistical significance in difference for both the neutral and angry’s angry value. The test results in Table 6.1 rejects the null hypothesis with a Chi-Square value of 6.125, which exceeds the critical value of 3.841. Therefore one can say that there is a statistical significance for the original hypothesis, with a P-value of 0.013, which in turn reflects the probability of the test sample having this tendency by pure randomness.

<table>
<thead>
<tr>
<th>no Null support</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Null support</td>
<td>0</td>
</tr>
<tr>
<td>Chi-square</td>
<td>6.125</td>
</tr>
<tr>
<td>Alpha</td>
<td>0.05</td>
</tr>
<tr>
<td>Critical value</td>
<td>3.841</td>
</tr>
<tr>
<td>P-value</td>
<td>0.013</td>
</tr>
</tbody>
</table>

Table 6.1: McNemar test on Hypothesis I
6.3.2 Hypothesis II: Users correctly identified the happy agent as more happy than the neutral one.

The data in Figure 6.2 shows no clear indication for this hypothesis to be true, as the happy agent scores lower than the neutral agent on the happy scale. The McNemar test in is also unable to reject the null hypothesis (see Table 6.2).

<table>
<thead>
<tr>
<th>no Null support</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Null support</td>
<td>4</td>
</tr>
<tr>
<td>Chi-Square</td>
<td>0.8</td>
</tr>
<tr>
<td>Alpha</td>
<td>0.05</td>
</tr>
<tr>
<td>Critical value</td>
<td>3.841</td>
</tr>
<tr>
<td>P-value</td>
<td>0.371</td>
</tr>
</tbody>
</table>

Table 6.2: McNemar test on Hypothesis II

There is a probability that the hypothesis failed because of the low semantic quality of the utterances as some users might have felt confused or rejected by the agents behavior. This is reflected in the gathered data as the neutral agent had overall higher scores in the perceived competency, intelligence and the modes overall clarity.

The happy agent was perceived as more polite than the other agents. This might be due to happy and polite utterances being easily mistaken for one another. The participants in the corpus gathering process might have had difficulties distinguishing these two tendencies and this might have lead to the happy dataset containing more polite phrases. The trained model would have then been influenced by these entries in its training set. A better screening and filtering of the training data might result in a better performance.

There was also a perceived issue in the training data gathering process of writing sentences which sounded happier than a neutral sentence. Angry proved to be an easier task as there seams to be a larger distance between sounding angry and neutral than in happy and neutral.

Another reason might be that the small sample size of 13 participants is not reflective of the overall population tendency and a larger or different sample might return a different result.
6.3.3 **Hypothesis III: Users enjoy interacting with an angry agents more than with neutral ones.**

This hypothesis was based on the fun to gruesome feedback. The data from Figure 6.1 already shows that the hypothesis can not be proven with the given data. The McNemar test results only verify that (seen in Table 6.3).

<table>
<thead>
<tr>
<th>no Null support</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Null support</td>
<td>3</td>
</tr>
<tr>
<td>Chi-Square</td>
<td>1.333</td>
</tr>
<tr>
<td>Alpha</td>
<td>0.05</td>
</tr>
<tr>
<td>Critical value</td>
<td>3.841</td>
</tr>
<tr>
<td>P-value</td>
<td>0.248</td>
</tr>
</tbody>
</table>

**Table 6.3: McNemar test on Hypothesis III**

It is relevant to note that most participants did find the angry mode more interesting than the other modes. A different or larger sample might result in a different conclusion.

6.3.4 **Hypothesis IV: Users enjoy interacting with a happy agents more than with neutral ones.**

This hypothesis was based on the fun to gruesome feedback. The hypothesis could not be proven as there is no indication of participants enjoying the happy game more than the neutral one. The happy agent received a slightly negative result (seen in Figure 6.2). The McNemar test can as expected not reject the null hypothesis (seen in Table 6.3).

<table>
<thead>
<tr>
<th>no Null support</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Null support</td>
<td>4</td>
</tr>
<tr>
<td>Chi-Square</td>
<td>0.8</td>
</tr>
<tr>
<td>Alpha</td>
<td>0.05</td>
</tr>
<tr>
<td>Critical value</td>
<td>3.841</td>
</tr>
<tr>
<td>P-value</td>
<td>0.371</td>
</tr>
</tbody>
</table>

**Table 6.4: McNemar test on Hypothesis IV**

The happy mode overall performed the worst on the interest and fun parameters. The approaches mentioned in *Hypothesis II* might also change the results of this hypothesis.
Chapter 7

Conclusion and Future Work

7.1 Conclusion

This project set out to discover new ways of generating emotional sounding texts from neutral sounding inputs, while keeping the meaning of the original input. It was decided after thorough research that a seq2seq model would be the most promising model for this project.

The most efficient seq2seq projects, Sutskever et al. [2014] and Jhamtani et al. [2017] used parallel corpora for training. After further research it was concluded that there was no parallel corpora for emotional utterances. Therefore the creation of a parallel corpus became a part of this project. An online form was created to allow voluntary contributions for three different emotions, angry, happy and sad. This approach only resulted in 8 contributions for the 20 entries long form, each entry was optional. The most entries where created for the angry and happy emotions which had a total number of about 165 entries, with addition of some embedding related entries.

The four seq2seq models all trained on the angry and happy datasets. Two of the four models used the RNN unit type LSTM and the other two used GRU, with each RNN type having one character embedded and the other having a word embedded model. The character embedded models overall performed better then the word embedded models, with the GRU models outperforming the LSTMs in almost all test cases. It is important to add that a word2vec approach to the word embedding might have returned better results as this approach is more commonly used for word embeddings. The decision to
use the one-hot embedding over the word2vec embedding was made to keep both the character and word embedding models more comparable as word2vec is not applicable to characters while the one-hot embedding is applicable to both.

The models were implemented into a game and 13 participants took part in an experiment to evaluate the models. The results of the experiment shows that the users could clearly identify the agent using the angry model, while happy could not be distinguished from neutral. There was also no conclusive tendency of participants enjoying the emotional models more then the neutral one. These results do not necessarily reflect the population’s opinion, the sample size was quite small, a larger or different sample might lead to different conclusions.

The overall results were positively surprising considering that the training corpus with about 165 entries was insignificantly small by most measurements, but it was still possible to generate models that could in some cases preserve meaning or convey the intended emotion (as the experiment results show), in some cases even both. The training and validation accuracies were also quite high for such a small dataset.

The model is quite promising, its weakest link is most likely the corpus. It is reasonable to assume that a larger corpus would result in a better overall performance and might also improve the user evaluation of the models.

7.2 Future Work

The project results are quite promising despite the before mentioned limitations (corpus and population sample size). The trained models show that the approach of using seq2seq models for emotion transfer might be applicable to real world challenges if further improved. This could be used in any system using conversational agents, like video games, virtual assistants (like Alexa [Papaioannou et al., 2017]) or customer service chatbots [Xu et al., 2017].

There are multiple ways in which the system might be improved upon. The first element to improve would be the creation of a larger and more general corpus as the current corpus is both small, which makes the training of the models less efficient and limited in its vocabulary which limits its capabilities if confronted with new words. It should
be possible to generate more data given more time and also use more commercial data gathering tools like Amazon Mechanical Turk [Niu and Bansal, 2018].

An alternative to the current models could be the use of a reinforced learning approach where there would be no need for a parallel corpus [Niu and Bansal, 2018], but these approaches are more difficult to implement and also produce worse results then the parallel corpus based models. It is still a path worth exploring.

There is also the option of implementing a model that could identify emotional phrases within utterances and replace them with a different emotion like the system created by Li et al. [2018]. This approach might not always return ideal results, but it could prove to be an efficient approach.

There are other areas where the system could be improved upon besides the model it’s using. The implemented voice synthesizer for the agent always has a neutral tone, even when the output is clearly angry. This could be improved upon with the use of the Speech Synthesis Markup Language (SSML) which is supported by many speech synthesizers including IBM-Watson [Sorin et al., 2017]. The implementation of SSML could further improve the user experience by making the agents more believable.

Another way of potentially increasing the users engagement and enjoyment is through the use of facial animations by the agent. This could be just lip-syncing or the capability of the agent of showing emotions [Valstar et al., 2016].

The models could also be implemented into a robot and it could be explored how emotional robots would change the behavior of the humans interacting with the it or if they enjoy interacting with an emotionally diverse robot more then with a friendly robot.

There are many areas in which the system could be further improved upon or used for.
Appendix A

Consent form
Speech Game Consent Form

Researchers:
Sibic, Ridvan A (ras10@hw.ac.uk),

Description:
The purpose of this study is to assess different game dialogue generation options and compare them to one another.

You will be presented with instructions on how to play the game, the scenario and the overall objective of the game. The test is not assessing your skills as a player nor the dialogue that you make, neither one will be judged or form part of this evaluation. You will be asked to complete a questionnaire after each game to help us assess your experience playing the game.

You are free to decline to participate in this study. You are free to end your participation at any time. All resulting data will be kept anonymous, should you decide to participate.

Voluntary consent:
I certify that I have read the preceding and understood its contents. Any questions I have pertaining to the research have been answered by the team. My signature below means that I freely agree to participate in this study and agree to the publication of the results for scientific purpose and to the distribution of the recording and transcripts of the sessions for research purposes so long as my identity is not revealed.

Participant ID: _______ Participant Signature: ________________________
Date: _______

Investigator’s certification:
I certify that I have explained to the above individual the nature, purpose, potential benefits, and risks associated with participating in this research study, have answered any questions that have been raised, and have witnessed the above signature.

Date: ________ Investigator Signature: ________________________
Appendix B

Instructions
Speech Game Instruction

Game scenario and Objectives
You are a travelling baker in a medieval town. You want to impress the locals in the new town you arrived in. You plan to do this by baking a cake, but you lack two ingredients for your cake:

- honey
- butter

Your goal is to find out where you can get the specific ingredients for your cake.

Game systems
You will play three different dialogues. The order in which you play these dialogues will be determined by the instructor and you will be asked to fill out a form after each game. You interact with the game only with your mouse. You will be given the option of selecting a response after each time the agent speaks. Your response will be one of multiple displayed sentences in the left bottom corner.

Instruction
1. Fill out the form with your personal data (age, etc.)
2. When you start the game, you will see an option window where you can select one of three different dialogues (Test A, Test B, Test C). Please select them in the order given to you by the instructor.
3. Once you selected one the dialogue will start.
4. After the agent says something you are given the option to select from a list of buttons on the screen what to respond.
5. You are encouraged to explore the different dialogue options available in each test.
6. Your goal is to find the given ingredients.
7. The bartender will let you now if you have found the locations of all the ingredients.
8. Fill out the given form after finishing the game.
9. Repeat the steps from 2. onward, but select the next test given to you by the instructor until you have done all three tests.
Appendix C

Questionnaire "before"
Sentiment Experiment - before

Please fill out the following fields.

Participation-ID

Your answer

Age

Choose

How often do you play video games?

never 1 2 3 4 5 daily

Have you played role playing games before?

Yes

No
Have you participated in any way in an experiment involving the character in the image above?

- Yes
- No

SUBMIT

Never submit passwords through Google Forms.

This content is neither created nor endorsed by Google. Report Abuse - Terms of Service - Additional Terms

Google Forms
Appendix D

Questionnaire ”after”
Sentiment Experiment - after

Please answer the following questions.

Please put in your Participation-ID.

Your answer

Which test did you just finish?

- Test A
- Test B
- Test C

How do you feel about your latest test experience? Please choose the following scores as intuitively as possible.

- Grusome

  1 2 3 4 5

- Fun

  1 2 3 4 5

- Boring

  1 2 3 4 5

- Interesting

- Easy

  1 2 3 4 5

- Difficult
<table>
<thead>
<tr>
<th>Sentiment</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>confusing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>rude</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>calm</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sad</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lifeless</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>incompetent</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>unintelligent</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

How do you feel about the character in this experiment? Please choose the following scores as intuitively as possible.

Did you find all ingredients? Did the game character congratulate?
Did you find all ingredients. Did the game character congratulate you?

- Yes
- No

Never submit passwords through Google Forms.

This content is neither created nor endorsed by Google. Report Abuse - Terms of Service - Additional Terms
Appendix E

Data gathering form
Emotional tendencies in texts

Hi, this is part of a MSc project, which tries to teach machines to convey emotions in text form.

Please rephrase the following statements into the given emotional settings. You can leave some out if you can’t think of anything. Please answer as many as you can. The sentences should be close to the meaning of the original. Also don’t worry about making mistakes, just try as best as you can and thank you.

**Example: "I watched a movie."**

**Happy:**
I enjoyed watching a great movie.

**Angry:**
I wasted my time watching a stupid movie.

**Sad:**
I watched a depressing movie.

**"I walked home."**

**Happy:**
Your answer

**Angry:**
Your answer

**Sad:**
Your answer
The rest of the form included the “I walked home.” section of the form, with the emotional fields, for each of the following (total of 20) utterances:

- I walked home.
- What do you want?
- What time is it?
- There is a town nearby.
- That's not what I want.
- Come here.
- Go away.
- Hello stranger. May I help you?
- I need your help.
- Do you know anything about that?
- I am sorry, I don’t know anything about that.
- Can I do anything for you?
- Here you go. Anything else?
- You are free to leave, but this is my house, so I do whatever I want!
- Here you go. A nice start for a long lasting friendships, cheers!
- Sorry, I am not looking for a relationship right now.
- There isn't much to tell, why don't we talk about you?
- Sure, give me a few minutes to get ready.
- Where can I find it?
- I guess I'll see you around, I hope you enjoyed our chat.
Bibliography


Google (2018). Google-forms. [https://www.google.co.uk/forms/about/](https://www.google.co.uk/forms/about/). [Online; accessed 8-August-2018].


Bibliography

Uphill Battles in Language Processing: Scaling Early Achievements to Robust Methods, pages 43–47.


