Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

Subdirección Académica

Sentiment Analysis through a chatbot

por

Alexander Espronceda Gómez

como requisito parcial para obtener el grado de

ingeniería en tecnología de software

Enero 2022

Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

Subdirección Académica

Sentiment Analysis through a chatbot

por

Alexander Espronceda Gómez

como requisito parcial para obtener el grado de

ingeniería en tecnología de software

Enero 2022

Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

Subdirección Académica

Los miembros del Comité de Tesis recomendamos que la Tesis «Sentiment Analysis through a chatbot», realizada por el alumno Alexander Espronceda Gómez, con número de matrícula 1742000, sea aceptada para su defensa como requisito parcial para obtener el grado de Ingeniería en Tecnología de Software.

El Comité de Tesis

Dra. Satu Elisa Schaeffer
Asesora

Dra. Sara Elena Garza Villarreal
Revisora

Dr. Romeo Sánchez Nigenda
Revisor

Vo. Bo.

Dr. Fernando Banda Muñoz
Subdirección Académica

San Nicolás de los Garza, Nuevo León, enero 2022

Contents

List of Figures

List of Tables

Agradecimientos

———(WORK IN PROGRESS)———

Resumen

Alexander Espronceda Gómez.

Candidato para obtener el grado de Ingeniería en Tecnología de Software.

Universidad Autónoma de Nuevo León.

Facultad de Ingeniería Mecánica y Eléctrica.

Título del estudio: Sentiment Analysis through a chatbot.

Número de páginas: 33.

Objetivos y método de estudio: En esta tesis se propone generar un software conversacional que interprete el texto introducido por un usuario y determinar su estado de ánimo, y reaccione de acuerdo con éste por medio de frases predeterminadas.

El método de estudio utilizado hará un análisis comprensivo de las redes neuronales, así como también de la comprensión suficiente de algo tan voluble y a veces impredecible como lo es la mente humana.

CONTRIBUCIONES Y CONLUSIONES: El algoritmo de entrenamiento utiliza un conjunto de datos específico para predecir lo más acertadamente posible qué está sintiendo una persona al momento de escribir alguna oración o frase.

Firma de la asesora: ______________________________
Dra. Satu Elisa Schaeffer

Chapter 1

Introduction

Human beings are social beings, this is widely known. To survive, we must band together and communicate with each other, bonding in the process. This is thanks to a neural process called *empathy*, which is defined as a three-part process that happens in our brains [11]. That happens roughly like this:

- Emotional simulation centered in the limbic system, which makes us mirror the emotional elements we're watching.
- Processing the perspective in the prefrontal and temporal cortex.
- Assessing the course of action to take, either showing compassion or doing something else. This is assumed to be based in the obitofrontal cortex, as well as several other parts of the brain.

This is clearly what is usually considered a human-only behavior, but there are studies that indicate that apes, dogs and rodents have been observed to take action at the presence of distress signals, either from humans or other members of their own species [16]. If this is true, theoretically, a machine could be taught to process signals of distress and react accordingly using a learning algorithm.

1.1 Justification

At first, the objective was to create an algorithm that could serve as a makeshift therapy chatbot that people could use when they were confused about their own feelings, but as time has passed, a lot of things have happened in my life regarding people with close-to-none empathy. This project could prove especially useful towards people who have trouble discerning when to console someone or having an idea of how other people or even themselves feel, such as the case of people with Asperger's Syndrome or other forms of high-functioning autism. To this end, the decision was made to work on this project.

1.2 Hypothesis

Empathy consists in a pattern of neurochemical reactions triggered by different situations. Machine learning could learn to identify these patterns. The hypotesis of this thesis is that machine learning could help people with a vague sense of empathy or self-knowledge to discern what they are feeling.

1.3 Objectives

In this section, the objectives proposed for this paper are established.

1.3.1 General Objectives

The objective of this project is to determine how the person that writes the input text is feeling according to the words in it. This could be achieved thanks to the technology present in machine learning algorithms and an extensive amount of datasets.

1.3.2 Specific Objectives

- Generating an algorithm capable of detecting key words related to mood in text.
- Predicting successfully the mood according to the input given.
- Giving feedback on the input, reinforcing it if positive or giving empathetic words if negative

1.4 Metodology

The tools that are used in this paper are mostly Python-based, such as TensorFlow, a neural network framework. This, combined with natural language processing tools and several filtering techniques will be used to achieve – or at least approach as close as possible to – the expected results.

1.5 Structure

The content in this thesis is divided in several chapters, each one of them talking about different information about either the topics that are relevant to the scope of this project or the general process that has happened to reach the goal.
In the second chapter, relevant concepts are discussed and expanded upon for better understanding of what this project's purpose. In the third chapter, existing literature is analyzed and compared to the present work, with comprehensive information and related concepts applied to each one of them.
In the fourth chapter, a general approach to the project's process is described.

Chapter 2

Background

Technology in the past decades has been advancing exponentially. So much, in fact, that we can relegate data analysis to them for better accuracy and reliability than what a human can possibly achieve. This is what is called as Machine Learning (sometimes referred only as ML). There is a variety of scenarios where it is useful, such as pattern recognition, which relates extensively to most of this project's work. In this chapter, some key concepts will be explained for easier comprehension of this thesis and the project itself as a whole.

2.1 Machine Learning

Machine learning can be described, broadly and figuratively speaking, as a black box where some data is inserted as an input and numbers come out of it as an output [20]. Some more advanced models of ML allow some internal parameters inside this figurative black box to be able to be tampered with, so that some characteristics of the input data can have effect on the output, these parameters are called *weights* [3]. Most ML algorithms have two stages: training and validation:

- Training processes the inputs and makes educated guesses, and in case of guessing incorrectly, depending on the obtained result, the weights are changed accordingly.

- Validation is as simple as it sounds, some input is fed to the algorithm and information needs to be compared to the real results to test the accuracy percentage.

One of these models that is one of the most used nowadays is the one called *Neural Network*.

2.1.1 Neural Network

A neural network works by using *neurons* that utilize layers that individually weigh the input given to them from the initial text or, if this has been processed already, from another neuron [3]. Likewise, similar to how biological brains work, these algorithms can only predict reliably if given enough data to train and validate their outputs with.

2.2 Sentiment Analysis

Sentiment Analysis (or Opinion Mining, as it is also known) as a tool for data analysis is arguably a recent happening. The term was coined in 2003 and has evolved ever since [12]. This type of data analysis has a lot of potential usages that have yet to be implemented in the daily life.

2.2.1 Concept

The specific execution of the algorithm varies depending on the intended purpose, but the concept and process that is used is generally the same:

- The sentence to analyze is broken down to its component parts, this process is called *tokenization*, and the resulting products are called, fittingly, *tokens*.
- Every token is then tagged, making it part of an internal dictionary or *lexicon*
- A score is assigned to every token depending on the used dataset.

The end score could be left as-is or can be reintroduced to the algorithm for a multi-layered approach depending on its focus [2].

2.2.2 TOKENIZING

Tokenizing is the process that happens while making tokens, the way it works is very straightforward: every word in the lexicon that a machine can read is assigned a number for easier reading. Taking the following example:

$$\boxed{\text{This is an example text}}$$

We can tell there are 6 words in the example phrase. So the tokenizing process would make the example look in the following way:

$$\boxed{1, 2, 3, 4, 5, 6}$$

where 1 corresponds to the word “This”, 2 corresponds to “is”, 3 to “an” and so on.

The interesting part about this process would happen if we used another example phrase, like the following:

$$\boxed{\text{This is another example}}$$

If we did the tokenization process, it would be processed in this way:

$\boxed{1,\ 2,\ 7,\ 4}$

Since the internal lexicon already knows some of the words in this second example, it reuses their token, adding new ones (in this example, "another" is 7) if needed.

This is fairly useful for a machine learning algorithm, since it will not have to compare such massive amount of characters in a string each time, and it would only need to evaluate integers. Whether its focus is either frequency or comparison.

Chapter 3

Related Work

The problem proposed in this thesis is not something new by a long stretch, since sentiment analysis was developed for this very purpose. There are many applications that already apply this kind of Machine Learning for several purposes. In this chapter, some related projects are listed and analyzed.

3.1 Related Projects

In this section, some literature is listed which proposes projects which have similar approaches to the present work, and some others that may not have the same objectives in mind but use algorithms that could be applied as well.

3.1.1 Similar Approaches

Blenn et al. [6] talk about three different text classificators with a focus on sentiment analysis from Twitter:

- Twitter Sentiment, which uses a Maximum Entropy algorithm[1].
- Tweet Sentiments, which uses Support Vector Machines[2] for classifications.
- Lingpipe, which uses both previous algorithms and also Naive Bayes[3]

Morris et al. [13] mention Koko, which uses the OpenAI API which is a counseling app for distressed teenagers in need of immediate psychological support, composed of a chatbot and sentiment analysis capabilities while Bird et al. [5] propose a chatbot developed to comprehend instructions, classifying them internally with a predefined bank of words, and reacting accordingly.

3.1.2 Sentiment Analysis in Other Areas

Pang et al. [15] draft out a movie review algorithm that was capable of detecting if the review was either positive or negative depending on the words used, and, Wang et al. [19] propose an algorithm that correlated the air pollution levels with the sentiment expressed in people's tweets. Capuano et al. [8] mention a hierarchical attention network to detect the polarity of a customer's review, with the added bonus of being capable of learning from new data. Chiril et al. [9] propose an algorithm that can detect hate speech in text using natural language text classification across several topics. Ahmad et al. [1] write about a classification system to detect if a tweet was deemed as extremist or non-extremist depending on the vocabulary used and a deep-learning algorithm. Similarly, Röchert et al. [17] report a Recurrent

[1]This algorithm works by having the bias that certain characteristics repeat more in certain categories in text. If no bias is found, the distribution is uniform. [14]

[2]Binary algorithm that can sort between two classes, or opt for classification in a "one-versus-everything else" basis. [18]

[3]This algorithm utilizes weights expressed in *-1*, *0*, or *+1* depending on the sensitivity of specific characteristics [4]. Works very similarly to a classic perceptron, which only uses *0* or *1*.

Neural Network that detect political statements in YouTube comments while also classifying them in *positive*, *negative*, or *other* depending on the topic.

3.2 Comparative Analysis

Since the projects included in this chapter are all focused in the same branch of algorithm, they have some concepts in common with each other and, in turn, with this project. Some of them are:

Machine Learning The type of algorithm needed for automatic processing, making the machine "learn" (hence the name) over time given enough data.

Neural Network A Machine Learning algorithm that uses weights and filters to output data.

Weights In ML, this is the name given to the internal value that a specific input has after being analyzed by the algorithm. With this, data classification can be achieved.

Text Processing Any type of algorithm that can understand text and output data based on its contents.

Natural Language Processing This is the method used for the algorithm to understand the content of the sentences, this is usually achieved by using tokenization but a preset corpus can also be used.

Sentiment Analysis This involves a ML algorithm, usually a Neural Network, that is able to analyze sentences and classify them according to the words used.

Corpus Preset internal dictionary that the algorithm uses.

Chatbot An algorithm that is able to reply to a prompt using natural language.

3.2.1 Opportunities for Improvement

One of the main positives of working with TensorFlow is the fact that it is a highly reusable code that can very much be ported to any system that can run Python.
It is important to mention GPT-3 as a whole, the framework that Koko – mentioned by Morris et al. [13] – uses is, to date, one of the most impressive AI algorithm to be developed, the downsides being that, being still in beta phase, is very resource-heavy, and its access is reserved to businesses through a fee, very expensive to use for the general public, especially students. That is why in this project, TensorFlow is used, which is free to use, does not need a lot of resources to work and has the advantages of being portable once trained, and also being easily modifiable if needed.

Table 3.1: Comparison between existing literature and the present work: ✓indicates the fulfillment of a criterion, otherwise × is used.

Project	**Neural Network**	**Text Processing**	**Sentiment Analysis**	**Chatbot**	**Open Source**
Blenn et al. [6] Maximum Entropy	✓	✓	✓	×	×
Blenn et al. [6] Support Vector Machines	✓	✓	✓	×	×
Blenn et al. [6] Lingpipe	✓	✓	✓	×	×
Morris et al. [13]	✓	✓	✓	✓	×
Bird et al. [5]	✓	✓	×	✓	✓
Pang et al. [15]	✓	✓	✓	×	✓
Ahmad et al. [1]	✓	✓	×	×	✓
Wang et al. [19]	✓	✓	✓	×	✓
Capuano et al. [8]	✓	✓	✓	×	×
Chiril et al. [9]	✓	✓	×	×	✓
Röchert et al. [17]	✓	✓	✓	×	✓
The present work	✓	✓	✓	✓	✓

Chapter 4

Project Design

The tools that are used in this paper are mostly Python-based, such as TensorFlow, a neural network framework. This, combined with natural language processing tools and several filtering techniques will be used to achieve – or at least approach as close as possible to – the expected results. Having all the concepts in mind, the proposed project has a major component which is the Machine Learning tools surrounded by several small modules such as the GUI and the chatbot components. As for the data used in this project, most of it comes from cleaned, classified tweets partitioned in training and testing datasets. In this chapter, the design of the project is explained.

4.1 Inner Workings Design

In this section, the data and the relations with the algorithm is explained, with a focus on the design itself.

4.1.1 Datasets

The datasets used in this project, as previously mentioned, consist in around 40,000 semi-clean, classified tweets in 13 categories [10], but as the scope for all of those labels exceeds the one proposed in this paper, the ones taken into consideration are as follows:

- Sadness
- Neutral
- Happiness
- Fun
- Worry
- Boredom

Even after eliminating non-critical labels, since the remaining labeled samples are not evenly distributed, leaving them as-is led to very inaccurate results, so a generalistic approach was opted for, classifying the end results in “Good”, “Neutral” and “Bad” depending on the overall wellness percieved from the input. This final filter works only with the training data, and works as follows:

- Sadness and Worry are in the “Bad” category.
- Neutral and Boredom are in the “Neutral” category.
- Happiness and Fun are in the “Good” category

4.1.2 Text Filtering

Since the chosen dataset is imported almost straight from Twitter with poor grammar, misplaced symbols, emojis and similar things, some cleanup has to be done to ensure peak performance.

- First, all text must be converted to lowercase.

- Then, all of the punctuation marks had to be discarded.
- After that, the stopwords[1] have to be omitted as well.
- Finally, for easier analysis, a process called stemming[2] is applied, so that all of the tenses of every verb are evaluated the same way while also avoiding corpus bloating.

These last processes were possible thanks to NLTK[3], which has its own repository of stopwords and stems. An example for this applied to data in the training dataset is as follows.

So sleepy again and it's not even that late. I fail once again.

Following the filtering order, first all the characters are converted to lowercase.

so sleepy again and it's not even that late. i fail once again.

After that, the text is stripped from all non-alphabetic characters.

so sleepy again and it s not even that late i fail once again

Next, all stopwords are culled from the sentence.

sleepy even late fail

The last step is to apply stemming to all the able remaining words, in this case, the adjective "sleepy" stems from sleep.

sleepi even late fail

[1] Words that are not vital for the sentence's meaning.

[2] Reducing a verb to its most basic components.

[3] Natural Language Toolkit, tool used specifically for these case scenarios. `https://www.nltk.org/`

4.1.3 Neural Network

For this project, as mentioned in Chapter 3, TensorFlow was opted for because of its characteristics such as being free to use, not needing a lot of resources to work and the advantages of being portable once trained. All of these traits are what makes this project unique and easily scalable. An LSTM Neural Network was opted for because of the increased accuracy that it offers compared to a regular Recurrent Neural Network.

Basically, LSTM is a subtype of a Recurrent Neural Network which has a certain amount of data be stored for longer periods of time so it can be used for future connections.

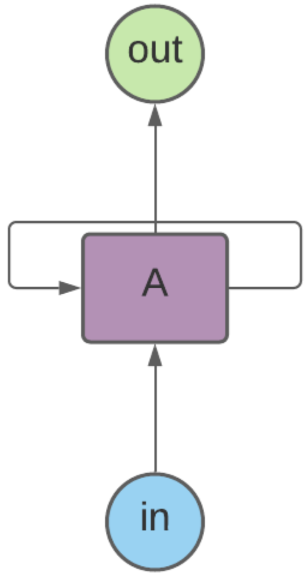

Figure 4.1: Basic Structure of a Recurrent Neural Network, where A represents the algorithm used.

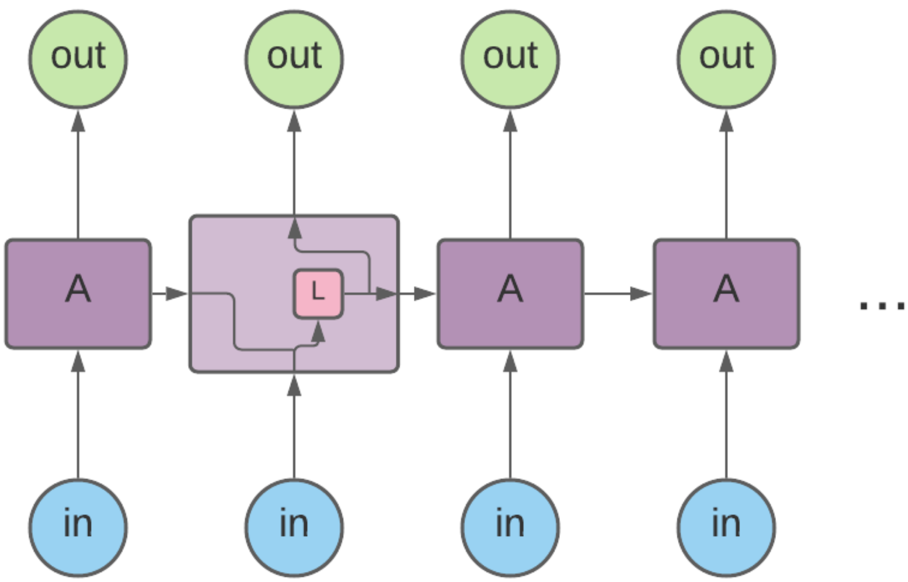

Figure 4.2: Structure inside an algorithm in a basic Recurrent Neural Network, where L represents the layers used.

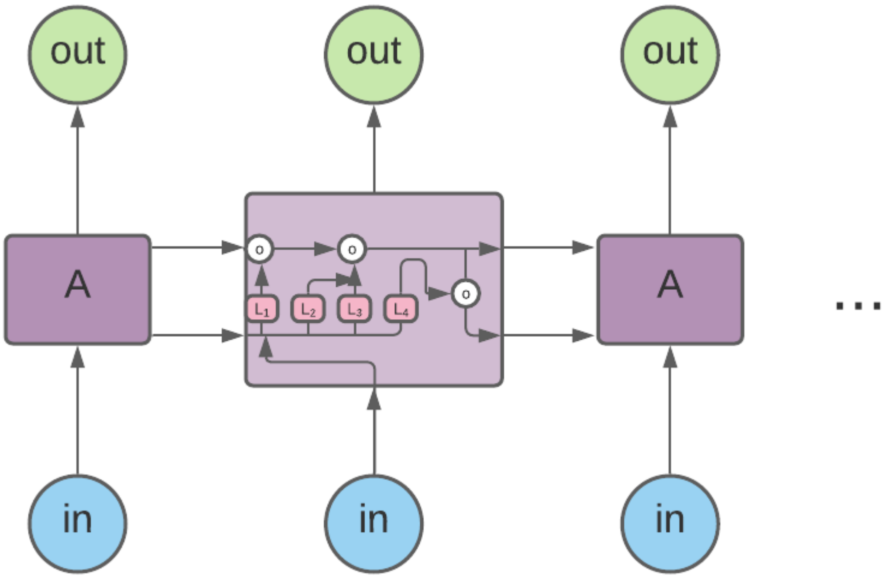

Figure 4.3: Structure inside an LSTM Neural Network, where the *o* represent the processing the data has when traveling from one layer to another.

4.2 Tools

This project is built on Python v3.8.10, The libraries used for this project to come to fruition are TensorFlow[4] v2.6.0 and Keras[5] v2.6.0 for the Neural Network section. Natural Language Toolkit[6] v3.5 (also known as NLTK) for the tokenization and stemming process. Chatterbot[7] for the chatbot's output. And, lastly, pygame[8] v1.9.5 for the GUI. Originally, *Ren'py*[9] was the chosen framework for this project's interface to work with, but – unfortunately for the proposed usage – it only works

[4]`https://www.tensorflow.org/`

[5]`https://keras.io/`

[6]`https://www.nltk.org/`

[7]`https://chatterbot.readthedocs.io/en/stable/`

[8]`https://www.pygame.org/news`

[9]An open-source Python framework focused mostly in the development of visual novels and other videogame formats. `https://www.renpy.org/`

with Python 2.7, which makes it incompatible with TensorFlow 2.0. Making a bridge between Python 2 and 3 would inevitably generate more issues that would take more time to solve, so it was scrapped in favor of the pygame library.

4.3 Inner Workings

In this section, I will highlight the most important parts of this project's code and their function. In case of needing further insight on the code used, the repository is online at `https://github.com/Alex-Ego/Affective-Computing-VN`.

4.3.1 Text Filtering

After the dataset has been properly located and ready to be used, the cleanup discussed earlier in this chapter happens in the following code snippet:

```
def tokenizing_process(message):
    # Pre-tokenizing
    tokens = word_tokenize(message)

    # Making them lowercase
    tokens = [w.lower() for w in tokens]

    # Filtering the punctuations
    table = str.maketrans('', '', string.punctuation)
    stripped = [w.translate(table) for w in tokens]

    # Filtering non-alphabetic characters
    words = [word for word in stripped if word.isalpha()]
```

```
    # Removing stopwords
    stop_words = set(stopwords.words('english'))
    words = [w for w in words if not w in stop_words]

    # Stemming words
    porter = PorterStemmer()
    stemmed = [porter.stem(word) for word in words]

    # Joining the resulting string
    message = " ".join(stemmed)
    return message
```

This works in the same way and order as specified earlier in this chapter.

4.3.2 Neural Network

After the text has been properly classified and ready-to-test, this is the structure of the neural network.

```
model = tf.keras.Sequential([
    layers.Embedding(input_dim=vocab_size,
                                  output_dim=embedding_dim,
                                  input_length=max_length),
    layers.SpatialDropout1D(0.15),
    layers.Bidirectional(layers.LSTM(32, dropout=0.15,
    recurrent_dropout=0.15)),
    layers.Dense(4, activation="softmax")
])
```

As the code shown indicates, this neural network has only 2 layers: LSTM for classification, and Dense for declaring that the result can be included in 4 categories, which include the three previously discussed categories, and one reserved for error/unknown purposes.

Chapter 5

Project Development

5.1 Datasets

There are several datasets on the internet, but none of them have the amount of sheer volume and actually useful data that is required for this task. The closest available was used, however, and it brought relatively acceptable levels of accuracy [7]. This dataset, paired with NLTK processing, stopwords and truncating words and verbs commonly used in the English language, was able to pinpoint if the input had a positive, neutral or a negative feeling about 40% of the time, approximately. This is not really a good number for such a small amount of labels, but it's an improvement nonetheless. Previous versions with different approaches, combination of layers and datasets had less than 20% of accuracy.

5.1.1 Pre-filtering

Since the dataset that was chosen was imported straight from Twitter with little to no filtering, some cleanup had to be done to ensure peak performance. The first problem was the punctuation marks, which were easy to filter out. The issues came after this with the so-called stopwords, which are words that do not really contribute to the overall meaning of the text. Luckily, NLTK[1] has its own repository of these words, so it was implemented. There was also an issue where verbs in different tenses were evaluated very differently, so a stemmer was implemented, which truncated words to its most basic features (aptly named stems) and prevented the loss to keep rising that much between epochs.

[1]Natural Language Toolkit, tool used specifically for these case scenarios. `https://www.nltk.org/`

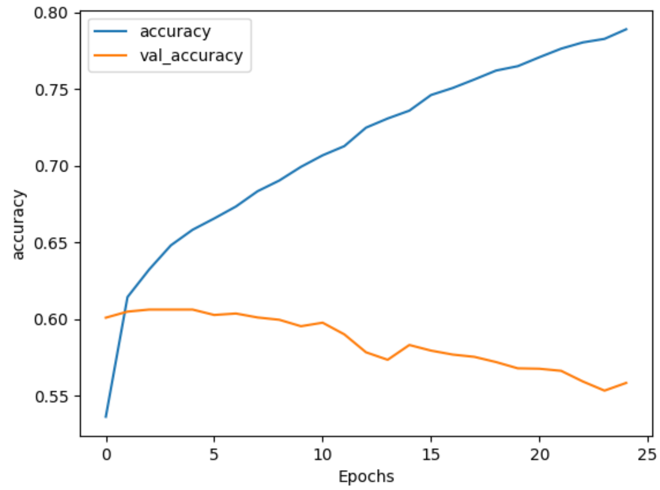

Figure 5.1: Accuracy Graph of the Algorithm Training on May 2020, with no NLTK stemming

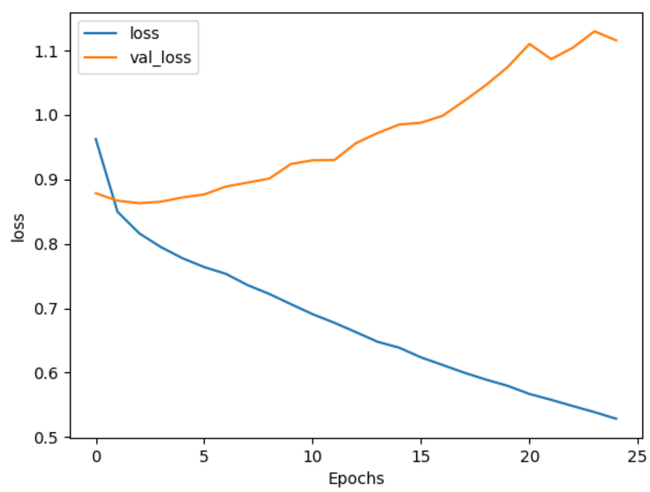

Figure 5.2: Loss Graph of the Algorithm Training on May 2020, with no NLTK stemming

5.1.2 Filtering

The dataset itself has several different sentiment labels to analyze, the ones being considered in the scope of this paper are:

- Sadness
- Neutral
- Happiness
- Fun
- Worry
- Boredom

But since they are not evenly distributed, leaving them as-is led to very inaccurate results, so a generalistic approach was opted for, classifying the end results in “Good”, “Neutral” and “Bad” depending on the overall wellness percieved from the input. This final filter works only with the training data, and works as follows:

- Sadness and Worry are in the “Bad” category.
- Neutral and Boredom are in the “Neutral” category.
- Happiness and Fun are in the “Good” category

Using a more complicated classification process would take an even amount of data in every category. Which, at the time of writing, no dataset readily available has.

5.2 Algorithm Used

A bidirectional LSTM algorithm was used with a softmax activation end layer. After much, much testing *rmsprop* was chosen as the optimizer because of its slightly better results overall. The internal lexicon is limited to 5000 items, and the maximum length of any given phrase after filtering is 30 characters. The training consists in 25 epochs on 75% of the dataset on a random arbitrary order, using the remaining 25% for validation instead.

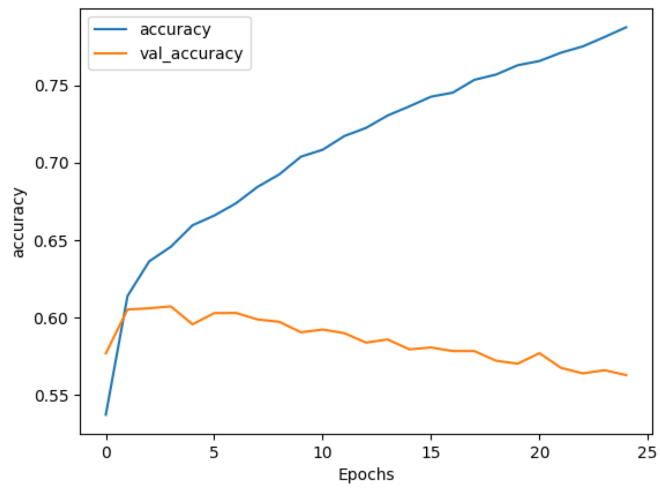

Figure 5.3: Accuracy Graph of the Algorithm Training on May 2021

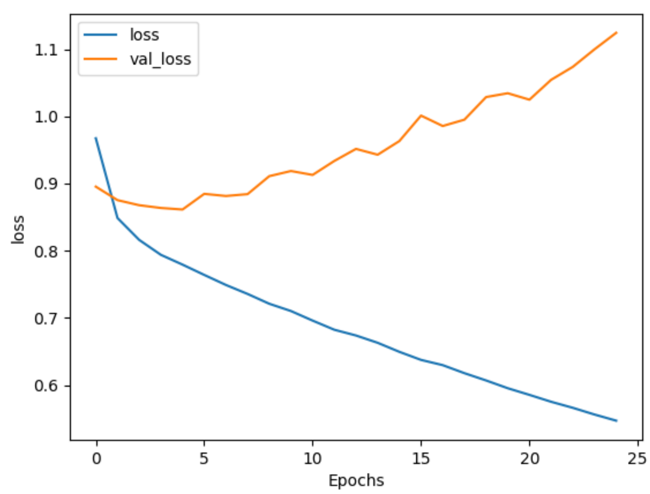

Figure 5.4: Loss Graph of the Algorithm Training on May 2021

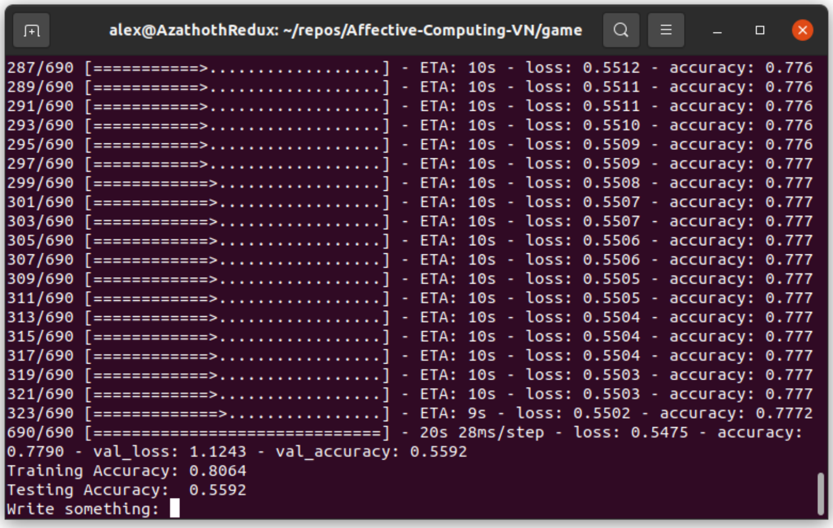

Figure 5.5: Debugging of the Trained Model

5.3 Interface

Originally, *Ren'py*[2] was the chosen framework for this project's interface to work with, but – unfortunately for the proposed usage – it only works with Python 2.7, which makes it incompatible with TensorFlow 2.0. Making a bridge between Python 2 and 3 would inevitably generate more issues that would take more time to solve, so it was scrapped in favor of the *pygame* library.

[2]An open-source Python framework focused mostly in the development of visual novels and other videogame formats. `https://www.renpy.org/`

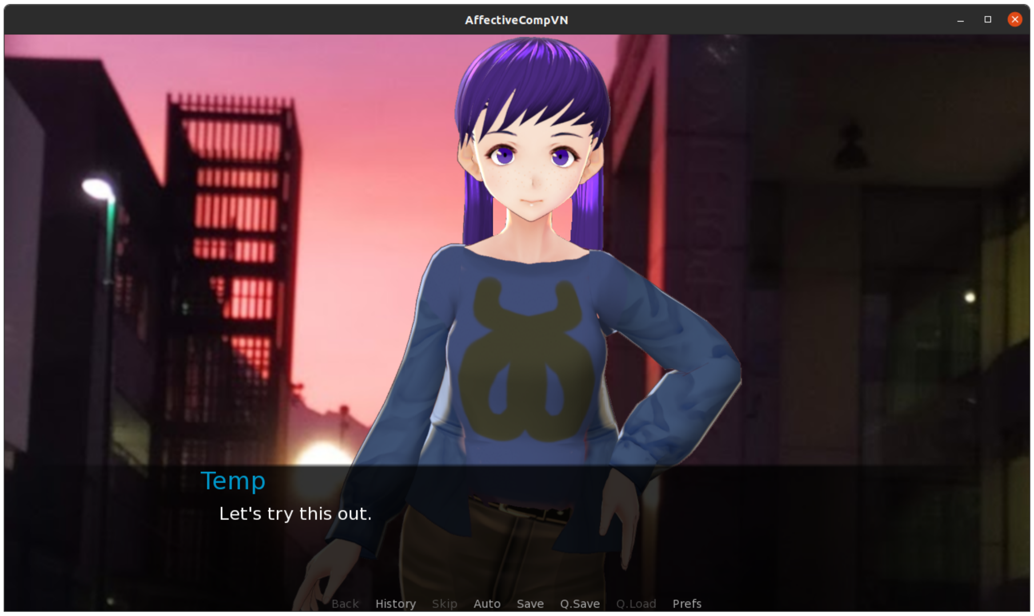

Figure 5.6: First version of the interface using Ren'py

Figure 5.7: Reacting positively to text in the "Good" category

The current interface is a hybrid between a Pygame screen, where the assistant appears to react to the input, and the console, where a person can input text to be analyzed.

5.3.1 Assistant

As for the character that is being used, it also has gone through some changes. Originally the idea was to make a low-poly character render to work with, but since 3D modeling-from-scratch skills exceed the scope of this paper, an alternative software was selected instead. Namely *VRoid*.

The main purpose for this assistant is to make people feel like it is her that they are talking to and not to some faceless machine, while also making it easier to the eyes. A more realistic, less animated style could have been used, but a friendly, less prone to uncanny valley approach to the design was opted for with this in mind.

Figure 5.8: First attempt at 3D modeling an assistant.

Figure 5.9: Assistant Ver. 2, now using VRoid.

Figure 5.10: Assistant Ver. 3, the current design.

Bibliography

[1] Shakeel Ahmad, Muhammad Zubair Asghar, Fahad M Alotaibi, and Irfanullah Awan. Detection and classification of social media-based extremist affiliations using sentiment analysis techniques. *Human-centric Computing and Information Sciences*, 9:1–23, 2019.

[2] Orestes Appel, Francisco Chiclana, and Jenny Carter. Main concepts, state of the art and future research questions in sentiment analysis. *Acta Polytechnica Hungarica*, 12:89–91, 2015.

[3] Taiwo Oladipupo Ayodele. Introduction to machine learning. *New Advances in Machine Learning*, Feb 2010. doi: 10.5772/9394. URL `https://www.intechopen.com/chapters/10703`.

[4] Daniel Berrar. Bayes' theorem and naive bayes classifier. *Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics; Elsevier Science Publisher: Amsterdam, The Netherlands*, pages 403–412, 2018.

[5] Jordan J Bird, Anikó Ekárt, and Diego R Faria. Chatbot interaction with artificial intelligence: human data augmentation with t5 and language transformer ensemble for text classification. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–16, 2021.

[6] Norbert Blenn, Kassandra Charalampidou, and Christian Doerr. Context-sensitive sentiment classification of short colloquial text. In Robert Bestak, Lukas Kencl, Li Erran Li, Joerg Widmer, and Hao Yin, editors, *NETWORK-*

ING 2012, pages 97–108, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. ISBN 978-3-642-30045-5.

[7] Fabio Calefato, Filippo Lanubile, and Nicole Novielli. Experimental datasets for sentiment analysis and emotion mining - emotion mining toolkit (emtk). Feb 2019. doi: 10.5281/zenodo.2575509. URL `https://doi.org/10.5281/zenodo.2575509`.

[8] Nicola Capuano, Luca Greco, Pierluigi Ritrovato, and Mario Vento. Sentiment analysis for customer relationship management: An incremental learning approach. *Applied Intelligence*, 51:3339–3352, 2021.

[9] Patricia Chiril, Endang Wahyu Pamungkas, Farah Benamara, Véronique Moriceau, and Viviana Patti. Emotionally informed hate speech detection: a multi-target perspective. *Cognitive Computation*, pages 1–31, 2021.

[10] CrowdFlower. Sentiment analysis: Emotion in text. Jul 2016. URL `https://data.world/crowdflower/sentiment-analysis-in-text`.

[11] Robert Elliott, Arthur C Bohart, Jeanne C Watson, and Leslie S Greenberg. Empathy. *Psychotherapy*, 48:1–2, 2011.

[12] Akshi Kumar and Mary Sebastian Teeja. Sentiment analysis: A perspective on its past, present and future. *International Journal of Intelligent Systems and Applications*, 4:2–4, 2012.

[13] Robert R Morris, Kareem Kouddous, Rohan Kshirsagar, and Stephen M Schueller. Towards an artificially empathic conversational agent for mental health applications: System design and user perceptions. *J Med Internet Res*, 20, Jun 2018. ISSN 1438-8871. doi: 10.2196/10148. URL `http://www.jmir.org/2018/6/e10148/`.

[14] Kamal Nigam, John Lafferty, and Andrew McCallum. Using maximum entropy for text classification. In *IJCAI-99 workshop on machine learning for information filtering*, volume 1, pages 61–67. Stockholom, Sweden, 1999.

[15] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up? sentiment classification using machine learning techniques. *arXiv preprint cs/0205070*, 2002.

[16] Stephanie D Preston and Frans de Waal. The communication of emotions and the possibility of empathy in animals. pages 2–3, 2002.

[17] Daniel Röchert, German Neubaum, and Stefan Stieglitz. Identifying political sentiments on youtube: A systematic comparison regarding the accuracy of recurrent neural network and machine learning models. pages 107–121, 2020.

[18] KP Soman, R Loganathan, and V Ajay. *Machine learning with SVM and other kernel methods*. PHI Learning Pvt. Ltd., 2009.

[19] Bingkun Wang, Ning Wang, and Zhongsheng Chen. Research on air quality forecast based on web text sentiment analysis. *Ecological Informatics*, 64, 2021. ISSN 1574-9541. doi: https://doi.org/10.1016/j.ecoinf.2021.101354. URL `https://www.sciencedirect.com/science/article/pii/S157495412100145X`.

[20] Xian-Da Zhang. Machine learning. In *A Matrix Algebra Approach to Artificial Intelligence*, pages 223–440. Springer, 2020.

Resumen autobiográfico

Alexander Espronceda Gómez

Candidato para obtener el grado de
Ingeniería en Tecnología de Software

Universidad Autónoma de Nuevo León
Facultad de Ingeniería Mecánica y Eléctrica

Tesis:
Sentiment Analysis through a chatbot

Nací el 17 de Noviembre de 1998 en Monterrey, Nuevo León, el mayor de los hijos de José Artemio Espronceda Estrada y Yadhira Lizet Gómez García.
Soy el primer hijo de la generación en mi familia, por lo que nunca sentí pertenecer, ya que mis tíos eran mucho más grandes que yo y mis primos mucho más pequeños. Por ello, siempre me encontraba pensando maneras de comunicarme con todos ellos "en su idioma" y lo lograba con relativo éxito. Pero a la persona que nunca pude entender fue a mi madre. Así que la mayoría de la inspiración de este proyecto se lo atribuyo a ella.
Me apasiona mucho el área de Análisis de Datos y Aprendizaje Máquina (Machine Learning), así como áreas como el Diseño de Videojuegos y la Psicología, por lo que este proyecto es la culminación entre mis pasiones más grandes para concluir la carrera de Ingeniería de Tecnología de Software.