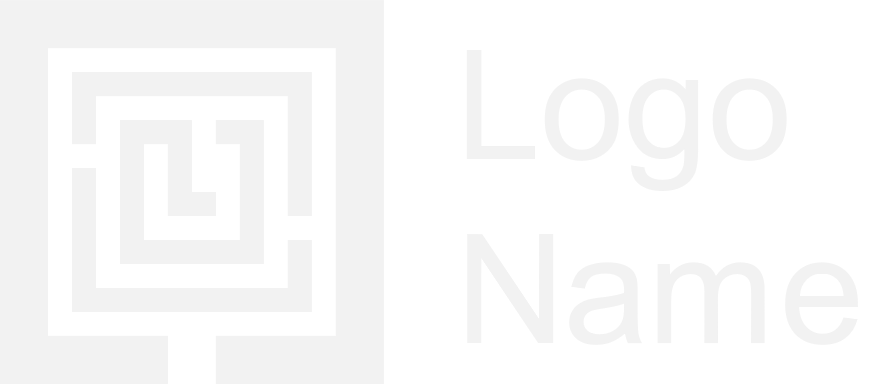


|  |
| --- |
|  |
| Fralysis  Freudian Dream Analysis Chatbot |
| 164870  PROJECT SUPERVISOR  Chris Thornton  Computing for Digital Media BSc  2018/2019 |



# Statement of Originality

# Acknowledgements

Project Supervisor: Chris Thornton

# Abstract

Mobile devices are our interface with the world. They enable us to shop, buy services, find help and recommendations. How we interact with computers is becoming increasingly sophisticated.

This paper investigates the feasibility of developing a mobile application that provides an insight into the meaning of an individual’s dreams based on Freudian dream analysis.

The project will:

* Evaluate current techniques used for natural language processing.
* Explore possible virtual environments to best simulate a therapy session.
* Document the process of developing a mobile application that provides Freudian dream analysis, covering:
  + Classifying user’s dreams from their input.
  + Provide a back-end application to perform dream analysis
  + Maintain a human aspect throughout the session

By the end of this report, you will have a basic understanding of Freud and his view on dreams, as well as an understanding of natural language processing. These ideas will be used to implement a mobile app named Fralysis which can conversate with a user and diagnose their dream.

Contents

[Statement of Originality 2](#_Toc6942641)

[Acknowledgements 2](#_Toc6942642)

[Abstract 2](#_Toc6942643)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1. Introduction1.1 Project Aim and Objectives The aim of my project is to simulate Freudian dream analysis using Fralysis – a mobile application. The analysis is focused on experiment, education and fun. Fralysis is not intended as a treatment for serious psychological issues.  A user will engage in a conversation with Fralysis. Over the course of the conversation, key elements of a user’s dream will be collected. The user’s dream will then be analysed using a store of historic dream data. The results of the analysis will be feed back to the user. Users can retrieve dreams from past session and compare them. They can then take this further with online research or seek professional guidance should they choose to. 1.2 Problem area and Needs of Intended Users There are many types of dreams [1]. Not all of them require analysis. 4 common types are:   * Normal dreams: a dream where the dreamer is not aware they are dreaming and where the experience of the reality of the dream does not provoke fear or anxiety. * Night terrors: a dream where the dreamer screams, experiences great fear, and flails while they are asleep. This type of dream is more common in children. * Night mares: a disturbing dream that is comprised of negative emotions, such as fear or anxiety. * Lucid dreams: a dream where the dreamer knows they are dreaming and can control the experiences within the dream.   A normal dream omitting any fear or anxiety poses no harm on the user. Dream analysis for these types of dreams would be for curiosity as opposed to resolving an unconscious confliction. Going to a therapist/dream-analysist would be a major investment for curiosity. This app would certainly appeal to this criterion.  The same can be said for people who experience lucid dreams. There may not be any lasting benefits of having lucid dreams analysed professionally, but this app can provide a fun, cheap experience for those who are curious.  For people who have experienced night terrors or nightmares and feel that interpreting these dreams will have significant benefits on their health could start with this app, but any real desire to unravel unconscious burdens should be taken to a qualified therapist.  As everyone has dreams of some kind, the intended user has an interest in the Freudian meaning behind their dreams. It should be usable to anyone.  The intended user so be provided with an easy to understand Freudian analysis of the dream, enabling the user to perform further reading if required. 1.3 Motivation The motivation for this project comes from an interest in psychology. This is a subject which effect everybody to some extent, especially in relation to dreams.  Developing an app that automates an otherwise expensive service will have the potential to widen the audience involved in this outdated, historic area. Although there are many dream interpretation apps, none of them target Freudian psychology with a focus on learning about his ideas and applying the latest natural language processing ideas. 2. Professional Considerations Although this project is intended for fun, it tightly relates to simulating therapy. Therefore, it will refer to the professional and ethical standards a clinic will follow.  My project aims to provide an external interpretation on a person’s dreams. Given the nature of Freudian psychology, this can disturb some people, especially if they are particularly suggestable.  Below I will go through the BCS Code of Conduct, and state how I will ensure my project meets the required guidelines.   2.1 Public Interest To maintain due regard for public health, privacy, security and wellbeing of others and the environment, I will have to make the purpose of the app very clear: to simulate the response a therapist may provide when interpreting one’s dreams in a fun, educational and experimental way.  These interpretations are by no means scientifically proven, they are filtered from many other interpretations surrounding a similar theme. This should not be taken as a conclusion but allow a user to take interest in the subject and decide whether they should follow up their concerns with a real therapist.  A cap on the sensitivity can be introduced. This would lead the application to become somewhat of an introduction to dream analysis. For example, a leading female character in the dream may refer to the Oedipus Complex. The interpretation should mention how this can translate into a representation of a mother, without going into exclusive details.  A major issue to consider is what information is stored from the user. As this app is responsible for comparing dreams from past sessions, a name is required. As this is only to reference a database, it can be made up.  However, the content of the dream is valuable in creating a database of anonymous dreams linked to Freudian meanings to create better results. This will be stated in the terms and conditions of the application.  There is a degree of social influence that can affect the results provided by the application. There is a large difference in how cultures interpret subjects. For example, eastern cultures are typically far more family orientated than the west, which has an autonomous approach. Freud’s ideas are western. This means these influences can make some interpretations invalid depending on the culture. This is something participant research should cover.  From background research, Google play store hosts several apps relating to dream interpretation from ‘Persian dream interpretation’ to ‘Biblical dreams’. These would provide different interpretations for the same stimulus. As my project is based around Freud, this will be the root of all interpretations. 2.2 Professional Competence and Integrity As this project can contains sensitive information, complete classification is guaranteed. No information outside of dream analysis will be kept on the participants. This is to comply with all legislations.  By evaluating multiple cultures, this aims to reduce any social biases that may offend users. Constant reference to the Freudian root of interpretation will ensure validity. These can be found in the appendix. 2.3 Duty to Relevant Authority The research I conduct will only relate to the subject area. This will be done with care and diligence. The ensure containment of my project, all research will be conducted on Sussex university campus under the circumstances mentioned above. 2.4 Duty to the Profession As this is a project which aims to promote interest in dream analysis, I aim to uphold its reputation by reflecting the outcomes of traditional therapy.  The reputation of the BSc will be considered and maintained throughout the project.  As this field has not exact answer, results should only stand as possible interpretations based previous diagnosis. Should any interpretation result in upset, this will be changed until there are no further issues. 2.5 Ethical Issues Validity of the success of this project can only be acknowledged using human test users. This will involve the participant playing with Fralysis. They will be informed that the primary purpose of this is not to leave an impact on their lives but refer to Freud’s ideas.  To comply with the ethical standards, it will be made clear that any interpretation of a dream (positive or conflicting) should only be received as a reference, and by no means a conclusion. This removes any harm or distressing thoughts to users.  To ensure the app remains ethical throughout, participants may be asked about what they find discouraging and what ideas should be left to further research if the user wishes to investigate further. 3. Related Work3.1 Dream Analysis The idea of dream analysis goes back to early civilisations, where dreams were associated to prophetic visions.  Freud’s ideas were thrown into fruition with his book ‘The Interpretation of Dream’ [40] in 1899. This introduced his unconventional ideas of Wish Fulfilment, Psychosexual Development, the Oedipus Complex, the Psyche and the Unconscious Mind [4].  Freud believed dreams were the ‘*royal road to the unconscious’* [2]and provide the best means of understanding the mind’s activities. Dream content has two forms: Manifest – literal object in the dream, and Latent – underlying meaning of symbol [3].  Symbols play a key part in Freuds ideas. He was against universal relationships between manifest and latent content: ‘*Every patient has their own personal interpretation’* [4].  The purpose of dream analysis is to determine the latent meaning and use this to address a patient’s concerns.  Freudian analysis is no longer mainstream, and a typical course of psychoanalysis can take many years. Therapy sessions involve 1 – to – 1 conversation between a patient and a therapist.  If dream analysis is to be implemented as a mobile app, an understanding of chatbots is required, and the limitations of implementing a historic idea outside the technological domain. 3.2 Chatbots The first major publishing of the idea of Chatbots was by Alan Turing in 1950’s, where he published an article: ‘Computer Machinery and Intelligence’ [5], which spawned the breakthrough idea of ‘The Turing test’.  The aim of the Turing test was to distinguish whether a person was either conversing with a person, or a computer. This idea gave birth to AI when trying to answer: ‘Can machines think?’.  Between 1960 – 2000 there were many chatbots with varying specialties from Natural language processing, therapist impersonations and simulating patients with mental health symptoms [6].  Joseph Weizenbaum developed one of the first chatbots [7.1]: ELIZA (1962) with the purpose of helping people explore feelings and emotions in a therapeutic (free association) fashion: ‘*it has a crucial psychological utility in that it serves the speaker to maintain his sense of being heard and understood*’ [7.2].  Although having some success as being interpreted as a human by some users, it did not pass the Turing test. It did leave its legacy: ‘*principles used in ELIZA laid a foundation for the structures of chatbots, such as keywords, specific phrases, and preprogrammed responses’* [7.2].  Contrary to ELIZA, PARRY (1972) [8.1] developed by Kenneth Colby was a chatbot which simulates a paranoid schizophrenic. Results showed that only 48% of psychiatrist correctly identified PARRY as a chatbot. PARRY and ELIZA can chat together, and the advancement in PARRY clearly affects results [8.2].  ALICE [9] succeeded ELIZA with better natural language capabilities and received three lorem awards. It still did not pass the Turing test.  Since the 21st century chatbot’s have progressed substantially, with big companies providing big investment. Sophisticated bots of our era include Apple’s Siri, Microsoft’s Cortana and Amazon’s Alexa. However, these are not developed for conversational purposes, but as agents of convenience – Play music, organize calendars and interface with the web.  Recently, Amazon have shown a large dominance in this field, with a desire to expand Alexa to a ‘socialbot’ [10]. This very much involves a machine learning approach, where the program is exposed to conversational data sets and human neural models of cognition and conversation [29].  The bot responds in a way that is most appropriate to the user’s message. This is done by finding patterns in the data set which resembles the context best.  As of now, there are still no chatbots which pass the Turing test [11]. If this were to be successful, it would be a significant advancement in AI.  Based on above factors and commercial needs, there is strong relevance for chatbots to be used, especially in therapy. From the preliminary examples, factors of successful chatbots involve:   * Efficient use of natural language processing. * Applied human characteristics. * Ability to recognize patterns.  3.3 Natural Language Processing Natural language processing and chat bot evolution are complimentary [12]. Data is taken and compared against a data set (Figure 1). An output is based on the best match.  Chatbots how chatbots work  Figure 1 – General NLP Model  responses are generated by using a similar idea to manifest and latent content [3]. The literals are identified by comparing the message to a database containing references to possible contexts.  For example, a user may ask a chatbot about ‘Donald Trump’. The database should have references to ‘Donald Trump’ as a ‘President’ and ’Businessman’. To give this meaning, context should be considered. This is done via a heuristic data store, which looks at the latent meaning regarding the conversation.  For example, ‘I really don’t like Donald Trump’ might have a response: ‘Yes. He seems to be a bad president and a bad businessman’. For the above response to be generated, an emotion – ‘really don’t like’, and direct object – ‘Donald Trump’ must be identified.  In the past, there have been two techniques to do this: Rule-based and Statistical [13]. Rule based traditionally uses nested if-else statements. Even on automating NLP, decision tree algorithms still produced complex conditional rule-based statements.  Statistical models reinvent this by finding the probability of a given context. By analyzing a corpus of business documents, references to Trump’s business ventures are likely to appear in a negative context [14].  From above, it’s possible extract a feature from a user’s input and find its emotion by looking at its dependencies. This is vital for dream analysis.  As mentioned in [13]: ‘*Two main issues that affect accuracy are unknown words and ambiguity’.* In chatbots, ambiguity is a product message structure. The advantage of using a chatbot is the ability to lead a user’s input. For this, sentence structure and patterns must be understood. 3.4 Sentence Structure Assuming we can successfully pre - process a user’s input with an NLP toolkit, there needs to be a process of identifying features within the sentence. This requires an understanding of sentence structure and common patterns which are likely to contain features that need to be extracted. 3.4.1 Sentence features In the context of dream analysis, the required features which need to be identified are: Subjects, Direct objects and Compliments [15]. 3.4.1.1 Subject The Subject of a sentence is defined as ‘*a noun phrase that functions as the topic of the sentence*’ [15].  For dream analysis, there are three subject classifications that can be made: The user themselves, someone they know or something meaningful like a car or pet. This means every dream will have a subject, regardless of how complex the sentence structure is. 3.4.1.2 Direct Object Direct Objects represent the manifest content of a dream.  For a Freudian analysis, a latent content must be attributed to identify the symbolism it represents in user’s unconscious [3].  For example: Little Hans [16] was a patient of Freud’s who had a fear of horses. After some analysis, Freud concluded that Little Hans was suffering from the Oedipus Complex, where the Horse resembled his father. Little Hans was afraid the horse would castrate him.  Using this example, we can identify a subject: Little Hans, a subject compliment: fear - giving a negative sentiment and the direct object: horse. In the given example, horse is symbolic of the Oedipus Complex.  In the context of dream analysis, a direct object in a dream should be any noun which references Freudian dream symbolism. In the example above, the horse is symbolic of the Oedipus complex. 3.4.1.3 Compliments Compliments are adjectives which provide sentiment to a sentence. These can either be directed at the subject or direct object.  For dream analysis, compliments hold information about the sentiment expressed by the subject towards the direct object, which can be applied to its symbolism. This means we can identify whether a symbol in the user’s dream is seem positive or negative context.  For a given sentence with a subject, verb and direct object, it can be said that: ‘the subject does the verb to the direct object.    Figure 2 – DisplaCy’s Dependency Visualization Example I  Figure 2 shows an example: The car hit the tree’. The subject is the car. The verb is hit. This is often used in a negative context. The direct object is the tree. We can re-phrase the above sentence to say: The subject shows a negative sentiment towards the direct object. This is fine for simple sentences with standalone verbs like hit.    Figure 3 – DisplaCy’s Dependency Visualization Example II  However, let’s take Figure 3: I was driving my nice car. The subject is the user, the verb is driving, and the direct object is car. Driving doesn’t provide a sentiment - it’s a neutral verb. The direct object has an object compliment – nice, describing the direct object in a positive context.  For dream analysis, identifying verbs and adjectives gives sentiment to the dream. This gives context to dream symbols. 3.4.2 Sentence Patterns With an understanding of the key components we need from a sentence, we can look at how this can be achieved. Going back to [15], there are 7 common sentence patterns which are likely to be exploited by the user to convey a dream.  The first pattern is known as ’The Be Pattern’, where sentences have the following structure:    Figure 4 –The Be Pattern  Figure 4 shows sentences start with a subject. This can either be a pronoun, proper noun or a noun. Pronouns which are most likely to appear in a user’s messages include: ‘I’, ‘I’m’, ‘we’, ‘he’ and ‘she’.  If a message begins with a first-person pronoun, it’s logical to assume the subject of the dream is the user.  If a message begins with a third-person pronoun, it probably isn’t the first message received from a user, but it does suggest the subject of that sentence is referring to someone else. To accurately identify if the subject of a dream is referring to someone else, we can identify if a proper noun is present, as its unusual to refer to yourself in third person.  This works in the case of a simple sentence structure like ‘The Be Pattern’ or ‘The Linking Verb Pattern’. However, it is possible that a ‘Transitive Verb Pattern’ is present Figure.    Figure 6 - The Transitive Verb + Two Complements  Verb Pattern  Figure 5 - The Basic Transitive Verb Pattern  For dream analysis, this gives the foundation of what information to extract from a sentence and how to classify it. It suggests that the implementation of the chatbot logic layer must recognize required features despite the message structure of the user. Furthermore, it suggests how to structure the chatbot’s responses to lead the user to provide specific information in their input. 3.5 Interface Design Considerations Mobile applications primarily take two inputs from users: strings and triggered events. Conversational UI [17] tries to simplify this further.  With the possibilities of NLP, there must be considerations into how best to interact with the user to simulating a therapy session as accurately as possible. This enhances data validity. Although chatbots have traditionally used a message interface [18], VR and robotics are the future of UI, and will likely produce different results.  For the given scenario: A user wishes to analyse their dream from the comfort of their home. Requiring the user to type inputs will certainly take way from the importance of free association [19]. Speech – to – text would be more suitable.  Furthermore, VR can be used to better immerse the user into the dream analysis. This would give better quality input data.  This isn’t always the case. Jeff Hawkins [20], founder of ‘Palm, inc.’ and now involved in applying neuroscience to computing, found ‘voice control is unsatisfactory and uncompelling’. Having users speak to their phone about sentimental information can be uncomforting and detract from its validity.  Considering the above, the app development stage will have a strong focus around User Centred Design and explore the best means of allowing a user to provide information to try draw a valid interpretation.  An MVC architecture means the same Freudian data model and NLP controller can be applied to various views. The preferred view for development would be through the terminal. The MVP could be a standard messenger interface to enable a user to converse with the chatbot. Given more time and technology, a VR domain can be implemented to focus on user immersion.  This architecture promotes privacy when using the app in various environments. Encapsulating the logic layer in a controller means you can easily update the NLP resources as they evolve. This is the same for the Freudian model, which can be expanded over time using machine learning. Testing confirms the controller is behaving as it should based on ideas from sections 3.3 - 3.4.  The output of the analysis will be a significant factor. Dreams are often ambiguous. The best method of conveying the post - processed information will be in a readable and descriptive string variable.  User centred design suggests using an MVC architecture. This enables multiple interfaces: terminal, MVP messenger, VR, etc. Just like the model and the controller, these can be updated separately and should still deliver the required functionality. 4. Background Research4.1 Evaluation of Existing Dream Analysis Applications4.1.1 iDream iDream [21] is an app for IOS which acts as an interactive dream dictionary. It allows a user to create dream journals. They can access the dream dictionary directly to enquire specific entities.  For example, going to see a comic and feeling envy for the quality of their jokes. The app provides an interpretation for a comedian: represents our childish instincts for a lack of seriousness. Envy can represent a jealous desire for something. iDream interface is shown in Figures 7 – 8.    Figure 7 – Dream Entity Dictionary  Figure 8 – Dream entity definition  Fralysis differs from this because it provides interactivity between the user and the Model. iDream uses a fixed lookup dictionary for analysis. Fralysis will improve on this by ‘Interviewing’ the user and using NLP to filter results. This enables personal responses, which is a requirement from [3]. 4.1.2 Dream-e Dream-e [22] takes a different approach from iDream. The app presents its self as a virtual therapist. The app looks like a standard messaging app. The users respond to questions presented by an A.I.  The app stresses it is not a dream dictionary, but a dream image analysis app. It will find out the therapeutic meaning of the images mentioned in your dream.  Dream-e is a generic dream analysis app. Fralysis aims to focus on Freudian psychology. Although Dream-e is a messaging interface, there is minimal human feel to its messages. Fralysis intends mimic a Freudian therapist and teach people about its ideas.      Figure 11 – Chat  Figure 10 – Dream Subject  Figure 9 – Dream entity definition    After experimenting with Dream-e (Figures 9 - 11), it was clear that the novelty of the chatbot was just to interface with the user to extract symbols of their dream and provide a bespoke dictionary definition. This is much like iDream.  Minimal natural language processing is involved. The app is not very satisfying. It does provide inspiration from a GUI perspective. 4.1.3 Morpheus Dream App Morpheus Dreams [23] is an upgrade over iDream with a quality interface. It primarily uses a dream dictionary to dynamically classify the main features of a dream.    Figure 13 – Dynamic entity definition  Figure 12 – Dream Journal  A clever feature is the implementation of the dream journal (Figure 12). As a user records their dream, key features are automatically interpreted (Figure 13). This is a clever use of natural language processing. It expands generic dream definitions by providing references to star signs and various superstitions.    Figure 14 – Morpheus’ capabilities  The app in general is more of a utility. It facilitates alarms and calendars (Figure 14) to motivate usage of the app and build a profile around the user.  If the functionality of Morpheus was implemented as a chatbot, this would be a great interface to Fralysis. This would enable past dreams to be compared and referenced in a therapy session. It paints a picture of what Fralysis should ideally be capable of doing.  Figure 15 – Dream App Comparison  Figure 15 compares current dream analysis apps on the market. It shows some of the features that should be attempted to compete with the current market. 4.1.4 Comparison Overall, the above apps approach dream analysis from different perspective.  All three apps approach dream analysis using the idea of dream dictionaries: iDream and Morpheus using dream journals. Dream-e uses a chatbot to interface with details of the users dream.  From a UI perspective, Dream-e seems to check out. it provides a viable approach to interacting with a user, although the logic used is not great.  The key difference with these applications and Fralysis is the idea of dream symbols. Fralysis should uphold Freudian values, and Freud felt strongly about the use of fixed dream symbols [39]: ‘*Freud undermined this naive theory of symbolic equivalences. All dream elements are ‘symbolic’, but they have private meanings that can only be discovered through the dreamer’s associations’*.  To overcome this, an extensive model will have to consider key Freudian ideas. The logical processing will involve simulating a therapy session as close as possible. There will be multiple levels of the interview process. Multiple dreams should be analyzed for patterns.  Fralysis analysis should associate the manifest dream content to a latent interpretation based on a Freudian model. This should be paired with an object compliment and targeted towards the subject of the dream. 5.2 Evaluation Languages and Packages There are many computing languages which have libraries specialising in areas which will be required for producing this app. Below will consider the certain aspects that must be implemented. This includes:   * Communicating in human – like fashion with the user. * Pre – processing of users input via a natural language toolkit. * Analysing features and sentiment of users input using sentence patterns. * GUI to implement messenger and allow user interaction.   Each section will contain a possible language to provide this. This is dependent on available toolkits. This section is structured so the most valuable language is considered last.  Android applications are typically developed in Java. 5.2.1 Java Java has a long running history. Initially developed for interactive television but struggled for being too advanced for cable transmission at the time.  Popularity came to Java when java applets became incorporated in web browsers via the java virtual machine. This meant java apps could run on any platform by compiling them through an intermediate representation called ‘java bytecode’. 5.2.1.1 Scanner Java has a class called Scanner which belongs in the util.java package. It can be used to parse primitive types and strings via regular expressions. Its breaks an input into tokens which is required for NLP.  Scanner will allow a user to enter a dream into console and is useful in extracting inputs at a development stage. Due to its long running history, it is a reliable library.    Figure 16 - Chatbot using Scanner  One limitation is that it is not very efficient. It doesn’t provide any text pre - processing. This brings in the need for a more sophisticated tool. 5.2.1.2 JavaFX Despite some saying it has become outdated, JavaFX is a key component of the Java SE environment. JavaFX was created to take over Swing, which was the original means for creating GUI for Java applications.  Image result for java swing  Figure 17 - Swing App  Swing class extends JFrame – the window that contains UI. JPanels are used to add components to the JFrame. JavaFX uses a theatrical approach of having a stage class which represents the screen, and various scene classes which handle the components making up the UI.  Image result for java fx  Figure 18 - JavaFX App  Java is multi-platform, and JavaFX combines the JDK and JRE on supported systems. Although there is no official Oracle distribution for JavaFX on Android, you can use the ‘JavaFX-Android SDK’ to create an Android project based on your JavaFX Application. 5.2.2 Python Python is one of the most popular languages (2018). It has a broad use, from desktop and web-application development to sophisticated scientific data analysis - Both of which are required for the project. 5.2.2.1 Pandas Pandas is an open source library used for data analysis and generating visual representations of this. Dataframe objects (Figure 19) are used for manipulating data.  C:\Users\Mr Hickes\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\AD3630C0.tmp  Figure 19 - Pandas Dataframe  Implementing this correctly helps support logic validity. Patterns are visually recognisable as shown in Figure 20. This can prove the dream analysis is working as expected.  Image result for pandas python  Figure 20 - Pandas Graph  If Fralysis were to expand from a chatbot, it could be used to provide visual analysis of a user’s dreams and which symbols and sentiments are recurring. Natural language toolkits For a general description of NLP, go to 3.4.  Python has well supported natural language processing libraries. These can be used for identifying features of a sentence.  Given Java’s age over Python, it has been in the NLP space for longer [24]. This section will focus on Python.  Using an NLP toolkit will save the time required to manually classify all word types that may appear in dreams. The two toolkits I will look at are NLTK and spaCy. 5.2.2.2 NLTK NLTK is a collection of programs and libraries for symbolic and statistical Natural Language Processing. It was developed to aid research and teaching in this field by providing empirical linguistics, cognitive science, information retrieval and machine learning – all relevant to the project.  The main features it supports are part-of-speech tagging and chunking.  Part-of-speech tagging involves marking up words in a sentence based on their definition and the context. For example, clowns are typically seen as scary to some people and funny to others.  NLTK provides the means to analyse words based on their context as opposed to stereotypes – which would be the biggest cause of misinterpreting dreams.  A limitation of NLTK is that it only supports the English language. It is also predominantly a research tool for study purposes, providing only a basic platform for prototyping and creating research systems.    For this reason, spaCy seems to be a better candidate. 5.2.2.3 SpaCy spaCy [26] is another opensource library that provides advanced natural language feature. It is designed for industrial strength word processing and is handled via a pipeline.  D1BC8CF6  Figure 21 - SpaCy Pipeline  The pipeline works by taking a text document and tokenizing it as illustrated in Figure 21. As documented by SpaCy [27], tokenization: ‘Segments text into words and punctuation and creates a Doc objects with the discovered segment boundaries’.    Figure 22 - sentence words to tokens  The tagger [28] gives words attributes. This describes properties like part of speech, lemma (Figure 22) and whether its representative of an entity – Person, organization, location, etc.  The tagger is initialized using a neural model (‘thinc.neural.Model’) which is essentially a data training file [29]: *‘A machine-learning algorithm is a mathematical model that learns to find patterns in the input that is fed to it. This input is referred to as training data’*.    Figure 23 - Using a data file to categories a token  Because the tagger is a machine learning algorithm, it is possible to replace the model and train your own tagger on specific data. If the application was further developed to support features like the history of Freud, it may be required to train the tagger on specific Freudian papers and books for more accurate associations. This can be done by changing the language data in Figure 23.  It’s fair to assume training an NLP pipeline on corpuses from the 19th and 20th century would certainly provide different results. The default model used by SpaCy [30] means: ‘*The parser and NER use an imitation learning objective to deliver accuracy in-line with the latest research systems, even when evaluated from raw text’*.  Fralysis requires name identification, as well as sentence features. SpaCy provides this through its part of speech attribute, where a name is typically associate to a proper noun. Sentence features mentioned in 3.4.1 can be extracted in a similar way.  Other domain advantages of spaCy is its support for multiple languages including German, Spanish, French, Italian and Dutch. As well as this it supports multi-language Named Entity Recognition. This is important for modern chatbots and improves user experience. 5.2.2.4 KIVY Kivy is GUI library for python. Although python has yet to facilitate application platforms, KIVY allows apps to be packaged for multiple platforms relatively easily.  As the app is a chatbot, the requirements from the GUI are straightforward: it must provide a chat screen where the therapist can send messages to the user, and the user can respond.  Using a MVC design means the model can be as complex as required, but ultimately a string is returned. Kivy role in the app is to provide a messenger view. A therapist message is written to the display. The user responds to this. The screen is updated, and the message is passed to the model to generate the next therapist response.  Image result for kivy diagram Image result for kivy diagram  Figure 24 – KIVY Flow Diagram  Figure 25 – KIVY architecture  Kivy allows for a minimal viable implementation for a working app. Ideally, this project could expand to explore different. For example, moving away from typing to speech-to-text input (Also supported with kivy).  Kivy is ideal for offline mobile use. The program can be packaged up for IOS, Android and Desktop.  However, given the nature of machine learning, file sizes can be very large. A solution would be to set the program up on a server and pass inputs via web-based interface. 5.2.2.5 HTML A web app could be used as a universal GUI. Html and php can be used to communicate with the python server. This still exploits the MVC model.  Strings are read from the web app view like that in Figure 26. These are passed to the backend logic controller. A string which represents the therapist message is returned to the view.  Image result for whatsapp web app  Figure 26 – Whatsapp Web App  This approach should be considered if expanding Fralysis. As offline usability is required for the MVP, KIVY is the library of choice for now. 5.3 Evaluating IDE environments IDE environments help organise projects and visualise interfaces. 5.3.1 Android Studios Android studios is a standard development platform for creating android applications.  Image result for android studios  Figure 27 – Android Studios  It has a plethora of libraries to structure the application layout. The frontend is typically written in XML.  A great benefit is visualising the interface dynamically. 5.3.2 PyCharm For Python development, PyCharm is the main IDE.    Image result for pycharm  Figure 28 - PyCharm  It resembles android studios and provides a good workflow between various classes and libraries.  PyCharm will most likely be used to develop Fralysis. 6. Requirements6.1 Functional Requirements Fralysis should have a messenger interface. It should prompt a user to converse with a therapist as if they were a patient. They should provide their name and describe their dream. If a name is recognised in the database, previous dreams of the user should be retrieved.  The controller should process a user’s message. If a name is provided or a dream is described, this should be classified correctly. The controller generates messages based on the classification. Therefore, responses from the therapist should resemble a human – typed message.  Classification is done by finding patterns in accordance to the Freudian model. The Model should contain at least 100 references to common manifest dream contents [3]. This should reference a possible latent content description for its Freudian meaning. It is also responsible for categorising compliments [15].  It should check to see if all required features have been collected and produce required requests for missing information.  Once an identified user has provided a complete dream, a choice should be given to provide more dreams or getting an analysis.  Analysis should start by referencing the subject of the dream. It will then provide an overall sentiment of the dream based on the subject compliment. It will then address which Freudian symbols have been classified. Identified manifest content is described using the latent content description found in the model.    A final reference to the Freudian argument should be provided to facilitate further research. 6.2 Non-Functional Requirements Responses should be generating within 2 seconds. This is to ensure the AI does not take excessive amounts of time providing responses. Shouldn’t be instantaneous – provide some realistic thinking time.  The app should be written in Python. The controller should be implemented using SpaCy. The interface should be implemented with KIVY.  The application should run on multiple mobile devices. This means packaging the app for IOS, Android and Windows. scaling should be automatic based on the device display.  To comply with professional conducts, explicit results should be filtered. 6.3 Condensed Requirements  |  |  |  | | --- | --- | --- | | Functional Requirements | | | | Ref | Description | Mandatory/Desirable | | F1 | Provide a textual - messaging application interface. | Mandatory | | F2 | Provide text-to-speech functionality. | Desirable | | F3 | Prompt user to provide name. | Mandatory | | F4 | Identify if user name is already in the database and retrieve previous user dreams. | Desirable | | F5 | Prompt users to describe their dream. | Mandatory | | F6 | Classify key features of a dream. | Mandatory | | F7 | Identify missing features from a dream. | Mandatory | | F8 | Identify a complete dream | Mandatory | | F9 | Provide analysis for a complete dream | Mandatory | | F10 | Suggest an application to real world situations. | Desirable | | F11 | Refer to original Freudian argument | Mandatory | | F12 | Output response in textual form | Mandatory | | F13 | Output response in auditory form | Desirable | | F14 | Provide a dream dictionary in Freudian style. | Desirable | | F15 | Visual construction of the dream. | Desirable | | F16 | Compare Dreams | Mandatory | | F17 | Android compatible | Mandatory | | F18 | IOS and Windows compatible | Desirable | | F19 | Handle ambiguous inputs | Mandatory | | F20 | Session should always terminate; does not have to be successful | Mandatory | | Non-Functional Requirements | | | | NF1 | Written in Python. | Mandatory | | NF2 | Product of TDD | Mandatory | | NF3 | Enforce MVC architecture | Mandatory | | NF4 | Support Android version up to 7.1.1 | Mandatory | | NF5 | Have a repository on Github | Mandatory | | NF6 | Provide at least 100 dream interpretations. | Desirable | | NF7 | Use SpaCy for NLP | Desirable | | NF8 | Use KIVY for GUI | Mandatory | | NF9 | Average response times within a second. | Desirable | | NF10 | A human therapist should be simulated. | Mandatory | | NF11 | Randomise messages | Mandatory |  Figure 29 – Requirements table6.4 Requirements Analysis The target audience for this project are interested in Freudian psychology and desire an insight into what their dreams represent in this context. This is implied through F5, F6, F10 and F11. An ideal system will ensure the result of F6 is factually accurate. It will also include an accurate representation of F14.  I will deem this project successful if the user is convinced they are receiving responses from a simulated Freudian therapist – NF10.  The app will be designed for smartphones. NF1 via NF8 and NF4 need to be fulfilled.  NF6 will determine the scope of the app and how detailed it is. The ideal product would contain an interpretation for every possible dream. This isn’t an achievable requirement as it would involve an excessive amount of storage and time. 100 interpretations provide an acceptable scope.  In summary, implementing the requirements will result in application capable of taking user inputs for a GUI view, determine a meaning based on the model and controller, and output a valid Freuds analysis. 2. Project Plan The project will begin by proposing an app related Freudian dream analysis - called ‘Fralysis’.  The first step involves getting a fundamental understand of the logic of the application. This is estimated to take 3 weeks.  Once knowledge on the topic is established, focus on the technical implementation will begin. Related apps will be evaluated. Three apps will be chosen which revolve around dream analysis. This is expected to take 2 weeks.  Implementing Fralysis will be the bulk of the project and will take all remaining time.  Freudian model needs to be created. This establishes core Freudian ideas that should be conveyed to the user in the analysis.  This will be followed by the logic of the app. This will begin by taking text from the terminal. SpaCy should pre – process this text  and responses are provided, this will allow for thorough testing of the logic of the app. The dream dictionary and libraries like SpaCy will be tested to ensure they produce the correct results.  The next stage will be linking an android compatible messenger interface to the logic. This would complete my first basic model of the app. After that, the focus will be on refining the outputs until they comply with all Freudian arguments.  TODO: \*\*\* LOOK AT EXAMPLE FOR ABOVE and relate to Gantt \*\*\* 6.1 Gantt Chart  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | 20/09 | 27/09 | 04/10 | 11/10 | 18/10 | 25/10 | 01/11 | 08/11 | 15/11 | 22/11 | 29/11 | 06/12 | 13/12 | | Initial thoughts |  |  |  |  |  |  |  |  |  |  |  |  |  | | Project Proposal |  |  |  |  |  |  |  |  |  |  |  |  |  | | Background / History |  |  |  |  |  |  |  |  |  |  |  |  |  | | Related Work |  |  |  |  |  |  |  |  |  |  |  |  |  | | Requirements |  |  |  |  |  |  |  |  |  |  |  |  |  | | Professional / Ethical |  |  |  |  |  |  |  |  |  |  |  |  |  | | Interim Report |  |  |  |  |  |  |  |  |  |  |  |  |  | | Participant investigation |  |  |  |  |  |  |  |  |  |  |  |  |  | | Coding |  |  |  |  |  |  |  |  |  |  |  |  |  | | User testing |  |  |  |  |  |  |  |  |  |  |  |  |  | | Evaluation |  |  |  |  |  |  |  |  |  |  |  |  |  | | Software Improvements |  |  |  |  |  |  |  |  |  |  |  |  |  | | Final report |  |  |  |  |  |  |  |  |  |  |  |  |  |  7. Design7.1 Logic From the background research, the look and function of the app can be designed. 7.1.1 Flow Diagram Figure 31 best illustrates the logic required to extract user dream information. It shows checkpoints needed for extracting information from a user.    Figure 31 – Flow Diagram  Fralysis should begin life in the home screen. This should provide button to navigate to the Dream Dictionary or the Dream Analysis Chatbot.  The Dream Dictionary provides an interface to the data model, where manifest object in a dream is coincided with a Freudian latent symbol definition [3].  The Dream Analysis page holds the chatbot logic.  The user is greeted. This simulates a therapy session introduction.  If a name has not been provided, request one. This simulates a real conversation with a stranger.  Request details about their dream. This searches for features in the input referred to in 3.4.1. The app should only proceed once the user has provided information on necessary features.  If the doesn’t provide the requested information, levels of repeat requests range from mild – ‘sorry, could you repeat that’, to urgent – ‘If you don’t tell me your name, I am going to have to kick you out’. This gives human appeal.  Once a dream is collected, the user should be asked if they wish to provide another dream or receive an analysis. Providing a new dream involves going back and requesting another dream. Getting an analysis gives feedback to the user and end the therapy session.  Encapsulating the logic like this enables code reuse. This is useful when transitioning from a standard chatbot to a more immersive VR environment.  7.1.2 Use case Diagram  Figure 32 shows which features are imperative from our session entities.    Figure 32 – Use Case Diagram  The required interactions are very minimal.  The user takes part in an interview, where they are asked to supply their name and give information about their dream; much like real therapy. Once the therapist has all the information they need, the users states whether they wish to provide another dream or receive their analysis.  The therapist has the responsibility of validating the user’s inputs. This is done by extracting features of their input and building up their dream until an analysis can be made.  Given these responsibilities, a class diagram can be designed. 7.1.3 Class Diagram Figure 33 gives a lower level view of the classes, methods and variables required in our program to achieve the required functionality.    Figure 33 – Class Diagram 7.1.3.1 GUI The KIVY library (reference in 5.2) implements GUI. This contains a Homepage Screen, Dream Dictionary Screen and a Dream Analysis Screen. These are controlled by a ScreenManager.  The user can talk to the chatbot by navigating to the Dream Analysis Screen from the Homepage Screen.  The GUI lets the user chat to the therapist via a messenger. This could be considered a container for the therapy session, which is hosted by the Therapist. 7.1.3.2 Patient A patient is the user.  A patient should have a name. This is how a user’s dream history is retrieved. It also provides a reference point in the chat.  Patients also have a list of dreams. These should be complete dreams and are used by the Therapist. 7.1.3.3 Dream A dream is defined much like a fictional story [31].  https://cdn-images-1.medium.com/max/1600/1*NefyLaPTXueKvHE3UkBa_Q.png  Figure 34 – Story Structure  Stories have a main character – this gives perspective. A main character is usually referred to in the subject of the sentence (3.4.1). This allows an interpretation to be directed in the context of the user.  The subject interacts with a direct object – noun. Take the film ‘Shrek’ [32]. Shrek is the main character, and the film is viewed from his perspective. The body of the story introduces a dragon. This is one of many direct objects in the film. The dragon initially symbolizes an antagonist – from preconceived fairytale knowledge.  The direct object elicits an emotion – direct object compliment.  A compliment is an adjective. This describes its nature. In Shrek, the dragon is described a ‘Fiery’ and ‘terrible’. This would lead to associating the direct object – Dragon, with a negative compliment.  Subject contexts are harder to identify. For example, ‘I jumped when I saw the terrible dragon’. The linking verb – jumped, shows a negative subject compliment. However, jumped is only negative because of the context.  Using this propositional logic, we can assume the subject compliment is the same as the object compliment and can be used to link the sentient of the dreams latent meaning to the subject.  With the features above, a dream can be given to a therapist, who can take its features and feed them back to the user.  Take the case of Little Hans [16] who had a phobia of horses. He thought horses would try castrate him. The subject is little Hans. The direct object is the horse. Horse has a negative compliment – castrate.  Through the therapy session, Freud associate the horse to Hans’ father, symbolizing the Oedipus complex. Bringing this to Hans’ conscious eradicated his phobia of horses. A similar case was found with his former patient – Anna O [33].  TODO: \*\*\*REFine this \*\*\*   7.1.3.4 Therapist The therapist controls the session and has Patients. Patients are identified by their name. If a patient has visited before, previous dream data should be retrieved and added to their list of dreams.  The therapist begins the therapy by starting an Interview with the user. 7.1.3.5 Interview The interview provides an interface for the flow diagram described in Figure 27. It controls when to move onto the next checkpoint.  Interview works by initiating one or more conversations and determines when the conversation cycle is complete – once the requested information is identified.  A complete interview is when there is no more information required for analysis.  The process of talking with the patient is handled in the conversation. Conversation The conversation writes messages to the GUI prompting responses from the user. The purpose of this class is to imitate a human therapist.  This is done by wrapping information requests in styled responses. This is straight forward for greeting a patient – ‘Hello there! How do you do?’ and requesting a name and dream in general.  However, requesting specific information in an indirect way requires therapist specific responses – ‘Lets develop your dream a bit more. How did the spider make you feel?’. The corresponding user’s reply will hopefully contain a subject compliment (3.4.1).  Another therapist styled request – ‘How did the spider appear in that context?’, is likely to lead a response containing an object compliment.  As there are likely to be multiple conversations in a session. Multiple message structures are provided for requesting features. This reduces the predictability of exactly what the therapist will reply with.  As well as message structures, there are request levels. These range from an initial request, to a repeat request and finally an urgent request. A message may not contain recognizable features. This gives the impression that the therapist is becoming frustrated with the user’s responses. It is also a tool to promote simpler user responses.  Assuming the user is giving valid responses, something has to classify the message to inform the conversation that a name or dream features have been identified. MessageTypeAnalysis If the user provides their name, a proper noun should be identified in their input. If it is, the name is returned to the conversation, passed back to the interview, and the clients name is set.  If the users input contains features of a dream, the entire message is returned to the conversation for further analysis. The input is stored in the dreams list of messages and given to AnalyseDream.  Before the message is returned, it is preprocessed by SpaCy language object and converted to a Document object. SetupSpacy Details on NLP implementation are provided in section 8.  SetupSpaCy is responsible for initializing the pipeline that prepares a user’s message for analysis.  It starts by taking a SpaCy language object and initialing it with a training data set as described in 3.3.  This is done using their ‘core\_web\_eng\_sm’ model. This turns words into tokens  TODO: \*\*\*\* Maybe this should in in implementation  As well as using their ‘core\_web\_eng\_sm’ model for named entity recognition and part of speech tagging, another model must be added to the pipeline to account for manifest and latent [4] dream content. Lexicon A lexicon is ‘*the vocabulary of a person, language, or branch of knowledge’.* Freudian specific terms – mother, house, can be added with a Freudian symbol – Oedipus complex, body. These can be given a Freudian description.  Multiple levels of analysis can be made like this.  The highest level references the dream context. References to symbolism’s and their sentiment follows. Finally, manifest content can be illustrated alongside its latent meaning [4] based on the lexicon.  This is added to the pipeline. If a user’s input matches to the lexicon, further analyzed by AnalyseDream.  AnalyseDream  This class takes a dream object. It looks at its latest message. Features are extracted and added to the dream.  If the dream does not have a subject, it will attempt to identify one (using rules from 3.4.1). Each dream should have one subject.  If a direct object is found that represents a manifest content [4], the latent symbol should be added to the dream’s topics.  If a compliment is found, this should be added to the dream’s ratings. Details on how this is done is provided in the section 8 of this report.  Once the dream object has been updated with the features found in the message, it needs to be validated. DreamValidator This assesses a dream. It takes the dream object and checks its contained features.  A list of missing features is generated and passed back to the conversation to be requested.  If there are no missing features, the dream is classed as valid.      By integrating the loosely coupled, highly cohesive classes above, a sequence diagram can confirm logical validity. 7.1.4 Sequence Diagram Figure 31 shows the low level interaction between objects in Figure 29 in order to implement the flow logic in Figure 27.    Figure 31 – Sequence Diagram  Using Figure 29, this gives a more detailed view on how the classes interact with each other, simulating the interaction between a therapist and a patient.  It illustrates the process of requesting a name as well as requesting and validating a dream. An iterative cycle is shown for repeating requests incase user information isn’t relevant. 7.2 User Interface The interface can be modeled to get an idea of how the app will look, and what complimenting colours compliment the idea of dream analysis. 7.2.1 High fidelity prototype. Figures 32 – 48 are prototypes showing how a user should interact with Fralysis. They illustrate how the final product should look and feel. Designs were created in Android Studios in XML on an Android VM.    Figure 32 – Introductory Message  Figure 34 – Greeting Response  Figure 33 – User Input      Figure 36 – Dream Description  Figure 35 – Name Response    Figure 37 – Dream Sentiment    Figure 40 – Completion  Figure 39 – Analysis  Figure 38 – Post Options  Figures 32 – 40 show the simplest conversation required to complete a therapy session.  Figures 45 -\_ show a possible means of comparing dreams.    Figure 43 – 2nd Conversation  Part 3: Positive Dream  Figure 42 – 2nd Conversation  Part 2: No Dream  Figure 41 – 2nd Conversation  Part 1: No Name    Figure 44 – 2nd Conversation  Part 4: Analysis  Figures 41 – 44 shows two dreams with the same symbols and the same sentiment. Figure 41 shows the consequence of an inappropriate name. Figure 42 shows an unrecognized dream.    Figure 46 – 3rd Conversation  Part 2: Analysis  Figure 45 – 3rd Conversation  Part 2: Negative Dream  Figures 45 – 46 show two dreams with the same symbols but opposing sentiments.    Figure 48 – 4th Conversation  Part 2: Analysis  Figure 47 – 4th Conversation  Part 2: Negative Dream  Figure 47 – 48 shows two dreams without matching symbols and different sentiments.  These prototypes show what an MVP should look like. The combination of pink and blue compliment a relaxed environment [34]. The interface is clean and simple. Its easily recognizable as a messenger and the icon of Freud shows the chatbots response.  With an idea of how to design Fralysis, the implementation process can begin. 8. Implementation8.1 Programming Language Python is the language used to develop Fralysis. PyCharm was used as the IDE. This decision was made based on Python’s Natural Language support. As Machine learning and natural language processing is very new, python has become the best supported for this field. Unfortunately, this meant sacrificing the GUI support provided in a language like Java. 8.2 Project structure An object orientated approach was required.    Figure 51 – Tests  Figure 50 – Assets  Figure 49 – Project Structure  Figure 49 shows the classes. These reflect Figure 29. Figure 50 Shows the assets: A frame for the home page, pictures of Freud and sounds for button pressing. 8.2 Model The statistical model used to initialize the NLP pipeline was SpaCy’s ‘en\_core\_web\_sm’ model [35]: ‘*English multi-task CNN trained on OntoNotes. Assigns context-specific token vectors, POS tags, dependency parse and named entities.’*  As well as this, a Freudian lexicon was used to extend the pipeline to account for Freudian topics and ratings.    Figure 52 – lexiconEntries.tsv Snippet  The file used to populate the lexicon is shown in Figure 52.  The first column contains the direct object or compliment. This is followed by its Freudian symbol. The third column gives a rating. The final column gives the latent meaning for the direct object. A compliment will have column two and four set to none  The origins of these definitions can be found in the appendix.  Ideally, the tsv. file should be replaced with an SQLite database. This would also serve the purpose of storing past patient session data.  This is a feature that would have been implemented with more time. 8.3 View     Figure 55 – InputOutput.py Snippet  Figure 54 – main.kv Snippet  Figure 53 – GUI.py Snippet    To keep the model separate from the view, Figure 53 contains the KIVY logic. Figure 54 contains the KIVY data. Figure 55 allows different views to be initialized.      Figure 56 – Terminal View    Figure 57 – KIVY View  In order to export the app to a mobile. Buildozer [36] was used.  Buildozer is a set of tools that enable python/kivy applications to be package for android.  Currently Buildozer can only be execute on Linux.  If successful, the app will be installed onto the phone and installation output will be displayed in the command window.  Unfortunately, issues occur while deploying the full ‘spacy’ version due to incompatible binaries.  04-24 18:11:08.607 14218 14238 I python : ImportError: dlopen failed: “/data/data/org.fralysis.fralysis/files/app/\_python\_bundle/site-packages/srsly/ujson/ujson.so” is 64-bit instead of 32-bit  04-24 18:11:08.607 14218 14238 I python : Python for android ended.  Spacy has the following dependencies:  spacy -> thinc -> srsly -> ujson  ujson contains ‘c’ components and c binary libraries.  The 64-bit binary is incompatible with the target Android phone.  Solution may involve development of a bespoke python for android recipe. 8.3 Controller The bulk of the program involves simulating therapy. Most of this is can be found in 7.1.3.  There were specific requirements Fralysis had to implement. Maintain Therapist characteristics The interview represents the therapist’s thought process.    Figure 58 – Interview: init Snippet  Figure 59 – ask\_for\_name Snippet  Figure 60 – ask\_for\_dream Snippet  The interview initiates one or more conversations and determines when the conversation cycle is complete – the required information has been collected.    Figure 61 – Conversation: init Snippet    Figure 63 – get\_dream Snippet  Figure 62 – get\_name Snippet  The conversation requests information from the client in a human like fashion.  Figure 62 shows requests from standard to urgent, corresponding to Figure 64.    Figure 64 – Therapist Responses  Figure 64 shows lists with different phrases for requesting certain features. Responses are random. If the app is used multiple times, there should be variation in the chat dynamics.  Using this highly cohesive implementation, the chatbot feels real. The interview is direct. It doesn’t deal with the complexities of users, but it is aware when the required information is collected. Extracting user’s Name from input The most significant implementation was converting a string provided by the user into a useful data. This begins with extracting the users name.  MessageTypeAnalysis has a method is\_name(). This iterates through the tokens in a sentence contained in a SpaCy Doc object until a token with a proper noun attribute is found.    Figure 65 – MessageTypeAnalysis: is\_name() Snippet  This works fine when given simple responses. When testing more convoluted phrases, words which are not nouns are interpreted as proper nouns based on the sentence structure. This has been a common issue presented on the SpaCy Github page [37].  This can be solved by loading SpaCy’s ‘en\_core\_web\_lg’ model instead of ‘en\_core\_web\_sm’. This model only recognized true names as being proper nouns.  However, the large model exceeds 800+ Mb, as opposed to 29Mb. This led to responses going from within a second to over 10 seconds, which breached requirement F\_.  Therefore, the compromise has been made to use the small model and assume the user keeps things simple. Most valid names are correctly identified. Extracting user’s Dream from input Recognizing dreams requires extending the pipeline to accommodate for Freudian specific terms. Just as the pipeline attaches part of speech and dependency attributes to tokens, it’s possible to add custom attributes.    Figure 66 – SetupSpacy: init() Snippet  Figure 66 shows attributes required from tokens relating to Freud. For example, tokens like ‘mother’ and ‘father’ are distinctive features to Freud, and visions of parents relates to the Oedipus / Electra complex.  As this idea is redundant in outside of Freud and in the 21st century, the default model doesn’t provide this data. A lexicon is defined as: ‘The vocabulary of a person, language, or branch of knowledge’.    Figure 67 – SetupSpacy: Lexicon Snippet  Figure 67 shows a class representing a dictionary data structure containing lexicon entries.    Figure 68 – SetupSpacy: LexiconEntry Snippet  Figure 68 shows a lexicon entry. It contains a lemma – the simplest form of a word.    Figure 70 – Constants: Rating Snippet  Snippet  List  Figure 69 – Constants: Topic Snippet  List  Topic associates a manifest content [4] with a symbol – ‘Oedipus complex’, ‘Psyche’, etc.  This allows multiple levels of analysis – instead of directly associating manifest content to Freudian interpretation. If the Oedipus complex is present multiple times, this should be feed back to the user.  A manifest content needs a sentiment. This makes it possible to identify whether the subject or topic is portrayed positively/negatively.  As seen from Figure 70, the Rating enum has 7 values. All adjectives should have a rating attached.  This alone is only capable of analyzing single adjectives. The idea of intensifiers, diminishers and negation must be introduced.    Figure 72 – Constants: Diminishers List  Figure 73 – Constants: Negation List  Figure 71 – Constants: Intensifiers List  Intensifiers increase the power of an adjective. Describing something as very good is more positive than just good. Fortunately, there are only a few common words that fit into this category, as shown in Figure 71. Common diminishers are given in Figure 72 and Negation in Figure 73.    Figure 74 – AnalyseDream: is\_essential() Snippet  When getting the topic and sentiment of a sentence, tokens are passed through a function called ‘is\_essential’ (Figure 74). If a rating token is found, ‘combine\_rating()’ (Figure 75) checks if there is anything modifying this token.    Figure 75 – AnalyseDream: combine\_rating() Snippet  Using the example ‘not very scary’, ‘scary’ is an adjective giving a very negative sentiment.  Combine\_rating() looks to the left and finds an intensifier present – ‘very’. This will increase the value of the adjective by one. ‘Scary’ has a value of -3, so signum will add -1 to this, leaving a value of -4. This exceeds the rating scale, so an internal method ‘\_ranged\_rating’ keeps the value within the possible range defined by the very positive / negative limits (-3 to 3).  A further examination finds negation – ‘Not’. The term ‘scary’ is very negative, but something being ‘not scary’ doesn’t make it very positive – It makes it somewhat positive. Figure 76 shows the negated map used to handle this condition.    Figure 76 – Constants: RATING\_TO\_NEGATED\_MAP Snippet  This isn’t a fixed mapping model as its debatable on the power negation has contexts.    Tokens with topics need latent descriptions (Figure 68). This resembles a dictionary that the referenced in the analysis.  Successfully implementing the Controller means the required data can be extracted from the user.  However, there are some vital issues to be illustrated. Freudian psychology is not a science: its pseudo – science, so topics given to Direct Objects are based on previous results from Freudian analysis. Psychoanalysis requires extensive details.  Freud also stated: ‘symbols in a dream are more personal, not universal’ [38]. Dream analysis is a personal analysis, and Freud was ‘cautious’ on defining symbols: ‘He did determine that there were some possibilities for universal symbols, such as poles, guns, and swords representing the penis.’  Giving horses an association to the Oedipus complex is fair in the case of Little Hans. However, horses can be interpreted indefinitely, and it is likely that a horse has been attributed to a completely different topic through a different psychoanalysis session.  To get a truly valid Freudian analysis of a dream, deeper machine learning is needed to replace the lexicon in Figure 67. By Training the pipeline on Freud’s books/research, you would extensively question the user until the key dream features resemble a past Freudian case.  The issue with implementing this is time constraints. It would involve converting analogue corpuses to digital. Even if you did this, attitudes and daily influencers are so different now that it would be hard to find connections between modern day dreams and dreams from the past due to technology and science, and life being so different.    For integrity sake, it’s worth re-iterating the true purpose of this project: provide a fun and experimental insight into Freudian psychology and dream analysis, with an aim to provide some history on some of Freuds ideas.  The way Fralysis aims to deal with this issue is by researching key topics. As found here: *’According to Freud, the number of things represented by symbols in dreams is not great: The human body, parents, children, siblings, birth, death, nakedness, and a few others*’ [39]. These are covered in the lexicon. 9. Testing9.1 Unit Testing   Figure 77 – Unit Test Structure  As Fralysis was a product of test-driven development []. Each class had specific purposes. Tests were written for individual functionalities, and the class was implemented until the tests passed. 9.2 Module testing     Figure 78 – Pytest Snippet  As well as Individual tests, cached data was used to automate conversations. These confirm the flow logic in 7.1.1 and are shown in Figure 9.3 User testing This involved giving the application to friends and requesting feedback from them.  10 friends were given the application to play with and answered five questions. Answers were answered quantitively: 1 = strongly disagree and 5 = strongly agree.   |  |  | | --- | --- | | Question | Average response | | My messages lead to an analysis being provided | 5/5 | | I was able to understand the analysis | 4/5 | | The analysis taught me something about Freud | 3/5 | | The chatbot felt like a human | 4/5 | | The chatbot gave the impression of being a therapist | 3/5 | | The conversation felt natural | 4/5 |   Figure 79 – User testing results  Figure 79 shows all 10 participants ended up with an analysis at the end of the session. This proves the flowchart in 7.1.1 was designed effectively.  From the analysis, most participants could decipher its meaning. This correlated with whether Freudian history was given.  Most participants thought the chatbot had human characteristics. When asking why, some participants replied: ‘it felt more real than Siri or Alexa in the way it answered questions’ . Whether it gave a therapist impression was mixed. This was due to enforcing the requirement of: ‘experiment and fun’. Most participants agreed that it felt like an ‘eccentric therapist’. Given the nature of Freud, you could class this as somewhat positive.  Going back to the flow of the conversation, most participants agreed it felt like a natural conversation. 10. Evaluation 10.1 Successes  The flow of the therapy session was successfully implemented.  An explanation of this can be due to an opportunity I had whilst developing this app. Whilst working for a games research company, I realized their process of evaluating qualitative data provided by players was very similar to my approach – find relevant references to the game, tag this and attach the players thoughts on it.  By substituting the Freudian lexicon for a game specific dictionary, I was able to test my feature extractor against my colleague’s analysis – who are professional researchers.  This led to a focus on the natural language processing aspects of Fralysis.  Due to confidentiality, I can’t provide details of data analysed, but it worked remarkably well in this context.  10.2 Limitations  These can be found at the end of each subsection in section 9.  The model should have been implemented as an SQLite database. This would have been done with more time. The tsv. file does suffice in providing a lexicon.  The KIVY view accomplished its purpose of providing a GUI and enabled for valid user testing.    Although it could be installed n a phone, the SpaCy libraries had to be removed. This meant it was installed without a model, and no functionality.  10.3 Overall  Overall, I have conducted a project revolving around Freudian dream analysis and considered the professional considerations whilst doing so.  As this is a very abstract, non-scientific subject, background research was required to understand what this project would exactly involve.  Types of dreams had to be classified and understanding why someone might desire to get them analysed.  Freuds ideas of dream analysis were overviewed. This provided the criteria required from a Freudian analysis. This looks at manifest and latent content as well as dream symbols.  The approach used to extract these features was natural language processing. This looks at what nlp means and considers sentence structures that need to be analyzed to achieve this. This was successfully achieved using Explosion AI’s SpaCy.  For the front end, possible interfaces were evaluated. As the implementation involves a chatbot, a standard messenger interface was used. This was successfully achieved using KIVY but didn’t couple with SpaCy.  Fralysis was implemented and given to users to test, who provided helpful feedback.  With more time, I would have resolved issues between SpaCy and KIVY to deploy a working model on mobile. I would also attempt to either expand the lexicon, or try a more machine learning approach. |
|  |
|  |

References

[1] Article: Types of Dreams

<https://www.huffingtonpost.ca/linda-lauren/dream-interpretation_b_4639124.html?guccounter=1&guce_referrer_us=aHR0cHM6Ly93d3cuZ29vZ2xlLmNvbS8&guce_referrer_cs=AEDTzZ5mGmBigU1L_Mj65A>

[2] Article: Freudian Dream Analysis

<https://en.wikipedia.org/wiki/Content_(Freudian_dream_analysis)>

[3] Article: Manifest and Latent dream content

<https://www.verywellmind.com/what-is-latent-content-2795330>

[4] Article: Freud’s Method for Interpreting Dreams

<https://www.freud.org.uk/learn/discover-psychoanalysis/the-interpretation-of-dreams/freuds-method-for-interpreting-dreams/>

[5] Academia: Computer Machinery and Intelligence

<https://academic.oup.com/mind/search-results?f_Authors=A.+M.+TURING>

[6] Article: History of Chatbots

<https://pcc.cs.byu.edu/2018/03/26/a-brief-history-of-chatbots/>

[7] Chatbot: ELIZA

1. <https://www.eclecticenergies.com/ego/eliza>

2. <https://www.masswerk.at/elizabot/>

[8] Chatbot: PARRY

1. <http://www.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/classics/parry/0.html>

2. <https://phrasee.co/parry-the-a-i-chatterbot-from-1972/>

[9] Chatbot: ALICE

<https://www.pandorabots.com/pandora/talk?botid=b8d616e35e36e881>

[10] Video: Socialbot

<https://www.youtube.com/watch?time_continue=13&v=9bSVpv3aoww>

[11] Article: Is the Turing Test Passed?

<http://isturingtestpassed.github.io/>

[12] Academia: Natural Language Processing: A Historical Review

<https://www.cl.cam.ac.uk/archive/ksj21/histdw4.pdf>

[13] Academia: Opinion mining

<https://thescipub.com/pdf/10.3844/ajassp.2017.843.851>

[14] Article: History of Donald Trump

<https://en.wikipedia.org/wiki/Business_career_of_Donald_Trump>

[15] Article: Sentence Patterns

<https://www.uhv.edu/student-success-center/resources/q-z/sentence-patterns/>

[16] Case Study: Little Hans

<https://www.simplypsychology.org/little-hans.html>

[17] Article: Introduction into Conversational UI

<https://rangle.io/blog/chatbots-an-introduction-to-conversational-ui/>

[18] Video: Chatbots and conversational interfaces

<https://www.youtube.com/watch?v=SvV57fuL_M0&t=787s>

[19] Article: Free Association

<https://www.goodtherapy.org/blog/psychpedia/free-association-in-therapy>

[20] Academia: Philosophy of the Hand Held

<https://books.google.co.uk/books?id=VvSoyqPBPbMC&pg=PA379&lpg=PA379&dq=jeff+hawkins+interface+design&source=bl&ots=d7MVSWkTTf&sig=diGK3omWxutgBwEP3bv3Erehz3g&hl=en&sa=X&ved=2ahUKEwjWor-IjcPeAhXDGewKHesDCdQQ6AEwAHoECAQQAQ#v=onepage&q=jeff%20hawkins%20interface%20design&f=false>

[21] App: iDream

<https://itunes.apple.com/gb/app/idream-dream-dictionary/id408076503?mt=8>

[22] App: DREAM – e

<http://technosophics.com/home/dream-e/Dream-e>

[23] App: Morpheus Dreams:

<http://morpheusdreamsapp.com>

[24] Article: Python vs Java for NLP

<https://stackoverflow.com/questions/22904025/java-or-python-for-natural-language-processing>

[25] Document: Stanford Core NLP

<https://stanfordnlp.github.io/CoreNLP/>

[26] Document: Spacy

<https://spacy.io/>

[27] API Document: SpaCy Tokenizer

<https://spacy.io/api/tokenizer>

[28] API Document: SpaCy Tagger

<https://spacy.io/api/tagger>

[29] Article: What is data training / testing sets

<https://www.quora.com/What-is-a-training-data-set-test-data-set-in-machine-learning-What-are-the-rules-for-selecting-them>

[30] Document: SpaCy Models

<https://spacy.io/models>

[31] Article: Fundamentals of a Fictional Story

<https://writingcooperative.com/fundamentals-of-story-structure-e199f131a891>

[32] Script: Shrek

<https://www.imsdb.com/scripts/Shrek.html>

[33] Case Study: Anna O

<https://www.simplypsychology.org/Sigmund-Freud.html#anna>

[34] Article: Complimentary colours

<https://brightside.me/article/the-ultimate-color-combinations-cheat-sheet-92405/>

[35] Github: SpaCy ‘en\_core\_web\_sm’ version 2.1.0

[https://github.com/explosion/spacy-models/releases//tag/en\_core\_web\_sm-2.1.0](https://github.com/explosion/spacy-models/releases/tag/en_core_web_sm-2.1.0)

[36] Documentation: Buildozer

<https://buildozer.readthedocs.io/en/latest/>

[37] Github: SpaCy name recognition issue

<https://github.com/explosion/spaCy/issues/2421>

[38] Article: Freud’s Dream Psychology & Analysis

<https://study.com/academy/lesson/freuds-dream-psychology-analysis.html>

[39] Article: Freudian symbols in our Dreams

<https://www.psychologytoday.com/gb/blog/out-the-ooze/201801/the-freudian-symbolism-in-your-dreams>

[40] Book: The Interpretation of Dreams

<https://www.freud.org.uk/learn/discover-psychoanalysis/the-interpretation-of-dreams/>

# Appendix

|  |  |  |
| --- | --- | --- |
| Symbol | Freudian Meaning | Origin |
| Human Body | The person is often represented in the form of a house | <https://www.psychologytoday.com/gb/blog/out-the-ooze/201801/the-freudian-symbolism-in-your-dreams> |
| Oedipus (Electra) Complex | describe a child’s feelings of desire for his or her opposite-sex parent and jealousy and anger toward his or her same-sex parent |
| Parents | Parents appear in dreams as kings, queens, or other highly respected persons |
| Children | children are symbolized as vermin |
| Siblings | siblings are symbolized as small animals |
| Birth | Birth is almost always represented by some reference to water. |
| Death | Death is replaced in dreams by taking a journey |
| Nakedness | nakedness is symbolized by clothing, especially uniforms |
| Sex Symbol | the great majority of symbols in dreams are sex symbols |

Appendix 1 – Freudian symbols

|  |  |  |
| --- | --- | --- |
| Word | Freudian interpretation | Origin |
| Mother | Parents appear in dreams as kings, queens, or other highly respected persons | https://www.psychologytoday.com/gb/blog/out-the-ooze/201801/the-freudian-symbolism-in-your-dreams |
| Father |
| House | The person as a whole is often represented in the form of a house – houses with smooth walls are men, and those with projections and balconies to which one might cling are women |
| King | Highly respected figures represent parents |
| queen |
| Horse | Horses were particularly suitable father-symbols because of their large penises | https://www.simplypsychology.org/little-hans.html |
| Sea | Freud argues that those experiencing an oceanic feeling as an adult are experiencing a preserved primitive ego-feeling | https://en.wikipedia.org/wiki/Oceanic\_feeling |
| Spider | spider in dreams is a symbol of the mother, but of the phallic mother, of whom we are afraid; so that the fear of spiders expresses dread of mother-incest and horror of the female genitals | http://www.freudfile.org/psychoanalysis/symbols.html |
| Water | Represents birth | https://www.psychologytoday.com/gb/blog/out-the-ooze/201801/the-freudian-symbolism-in-your-dreams |
| Seed |
| Journey | Journeys represent death |
| Naked | Represents the body |
| Three | Three is a symbolic substitute for the entire male genital |
| Bath | Represent nakedness |
| Tree | The penis alone is represented by long and upright objects such as sticks, umbrellas, poles, trees |
| Fountain | male genitalia are objects out of which water flows, such as faucets and fountains |
| Waterfall |
| Train |  |
| Car |  |
| Ship | female genitals are symbolically represented by objects which enclose a space capable of being filled by something |
| Flying |  |
| Snake | less obvious male symbols include reptiles, especially snakes, and fish, hats, and coats. |
| Fish |
| Mouth | female genitals are represented by objects which enclose a space capable of being filled by something |
| Church |
| Bottle |
| Box |
| Forest |  |
| Sunset |  |
| Mountain |  |
| Falling |  |
| Sweets | sweets frequently stand in for sexual delight |
| Plane |  |
| Bridge |  |
| Sword | objects that can penetrate the body and cause injury – think of knives, daggers, lances, swords, and firearms (especially revolvers) |
| Gun |
| Rocket | penis raises itself against the force of gravity results in symbolic representation by balloons, airplanes, missiles, and rockets |
| Island | area of your unconsciousness hidden away from the world | https://dream-meaning.net/place/island-dream-interpretation/ |
| Desert | lack of nourishing ideas and energy in your life | http://dreamstop.com/desert-dream-symbol/ |
| White | New experience and feelings | http://www.dreamgate.com/dream/hoss/ |
| Black | The unconscious realm. Moving into darkness = suppression, “death of the ego” (first stage of transformation). Beautiful shiny black = a positive view of the unconscious from which a new self and new potentials emerge |
| Blue | eel tranquil, peaceful and content |
| Red | feel intense, vital or animated |
| Yellow | joy and optimism |
| Green | establish myself, my self-esteem, my independence |
| Silver | feminine qualities and also how we interact with others |
| Gold | heart, compassion, persistence, and patience |
| Orange | expand my interests and develop new activities |
| Pink | feel romantic or loving toward someone or something |
| Purple | feeling erotic and intimate |
|  | After all, as Freud himself once said, “sometimes a cigar is just a cigar.” |  |

Appendix 2 – Freudian Interpretation

|  |  |
| --- | --- |
| Figure | Origin |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |