

Computational Creativity in Psychogeographical Activities: exploring a potential use case for computationally generated poetry



UNIVERSITY OF
LINCOLN

Bailey Symonds
SYM16653220

16653220@lincoln.ac.uk

School of Computer Science
College of Science
University of Lincoln

Submitted in partial fulfilment of the requirements for the
degree of BSc (Hons) Computer Science

Supervisor: Dr Jussi Holopainen

May 2021

Word count: 11,134

Contents

Contents	i
Acknowledgements.....	iii
Abstract.....	iv
Introduction.....	1
Literature review.....	3
2.1 Bespoke interactions	3
2.2 Recommender systems.....	4
2.3 Text generation	4
2.4 Psychogeography and Dérive.....	7
2.5 Summary	8
Methodology.....	9
3.1 Project management.....	9
3.2 Prototype development	11
3.3 Research methods	12
Design.....	14
4.1 Exploration.....	14
4.2 Chosen concept	20
Development.....	24
5.1 Features to include	24
5.2 Directions and prompts	24
5.3 Using users' answers.....	25
5.4 Prompting GPT-3.....	26
Evaluation	36
6.1 Overall experience	38
6.2 Individual tasks	38

6.3	Reactions to the output poetry	38
6.4	Relating to the poetry	39
6.5	Offering more poetry	39
6.6	Browsing previous poetry	39
6.7	Awareness of computational generation	39
6.8	Showing poetry versus not.....	40
6.9	Computational versus human poetry	40
	Discussion.....	41
	Conclusion	43
	Reflective analysis	44
	References.....	46
	Appendix.....	53

Acknowledgements

There are a few people who have been integral to this project's completion and success, and I would like to thank them here.

My supervisor, Dr Jussi Holopainen, for his constant enthusiasm during both this project and the summer project that inspired it.

My partner, Daisy, for her unwavering support and optimism.

My parents, for their continuous encouragement and generosity.

Finally, I extend my thanks to the participants who kindly volunteered their time to use and give feedback on the app developed in this project.

Abstract

As users seek more tailored content and computers' natural language processing skills improve, it is not unreasonable to assume a crossover in which computational creativity is used to meet the demands of the media-consuming public. Recommender systems are already used to offer relevant media to users based on their previous experiences, while artificially intelligent chatbots are well known and widespread across the web. This project develops a mobile application that guides users on a walk and asks them questions about their surroundings. Users' answers are passed to GPT-3 to generate relevant poetry, which is then displayed to the user. User interviews were undertaken and their findings analysed, suggesting users are keen to interact with computational creativity, in this case agreeing that its use is more enjoyable than the human-created alternative.

Chapter 1

Introduction

Computational text generation (a branch of natural language processing, itself a branch of artificial intelligence) focuses on generating texts in natural languages. Over the past half century, the quality of computationally generated text has improved significantly, with what is arguably today's best language model, OpenAI's GPT-3 (Brown et al., 2020), using deep learning, a form of machine learning inspired by neural circuits found in animal (including human) brains.

Many systems across the web utilise recommender systems to suggest content to individual users, bespoke to their previously shown interests and preferences. Following the trend of artificial intelligence's increasing computational power and its use becoming more and more common place, it is not unreasonable to imagine systems not only recommending bespoke collections of content, but also generating the content they recommend.

Though its computational techniques are relatively modern, text generation has been practiced for centuries. In 1305, Ramon Llull published *Ars Magna*, which described a method of generating answers to readers' questions, while in 1651, Georg Phillip Harsdörffer developed *Fünffacher Denckring der Teutschen Sprache*, a mechanism which could generate any German word (Schäfer, 2006).

Previous projects have combined computational text generation with bespoke interactions to create bespoke natural language text based on a user's bespoke input. For example, Rajcic and McCormack's (2020) Mirror Ritual generates poetry based on their system's perception of a user's current mood, and Shi's (2019) system that generates captions based on keywords extracted from images being posted on social media.

The project documented in this thesis aims to explore the design space for potential use cases of computational text generation. It aims to develop a

prototype mobile application to showcase one scenario in which computationally generated poetry could be used. The use case developed in this project is a mobile app which guides users on a walk, collecting their answers to questions about their environment, before generating poetry relevant to their walk. To do this, existing literature must be explored to get a sense of the design space. Speculative concept designs (Auger, 2013) must be created and evaluated, in order to select a single concept suitable for prototyping. First, requirements will be decided. These requirements will then influence a process of sketching potential designs, which will then be described in more detail. The most suitable concept will be chosen and developed further. Once this concept has been further explored and designed, it must be developed into a usable mobile app.

The project also aims to evaluate the suitability of computationally generated poetry in this particular use context. Potential users must be given the prototype app to experience the designed interactions. These users will then be interviewed about the prototype app and their experience while interacting with it. The findings and feedback gathered from these interviews will then be used to evaluate the prototype app and to aid discussion and reflection on the potential uses of computationally generated poetry in user-facing contexts.

Chapter 2

Literature review

This literature review will discuss the concept of bespoke interactions in regard to media consumption and current ways media is recommended to users. It then discusses the methods and history of text generation, as well as its usage and ability. Psychogeography, the ideology behind this project's prototype app, is also described.

2.1 Bespoke interactions

Smart assistants are already offering bespoke experiences to users with features such as Voice Match, Google's assistants' ability to recognise and distinguish between multiple users using the same device (Lyons, 2020). Voice Match allows the system to change its settings and information sources depending on who it is communicating with. Apple's Siri (Apple, 2021) and Amazon's Alexa (Amazon, n.d.) also use such features to give their users personalised experiences, showing the ability to have bespoke, personal interactions with smart assistants is a feature that many users desire.

“Leading companies are exploring ways to create bespoke experiences using human-centered design and digital technology to create personalized, enriched interactions at scale” (Duffield, 2021).

Explaining their predicted “bespoke for billions” trend, Duffield explains that major organisations have interests in creating media and experience bespoke for individual users, suggesting a cross-industry shift in focus from creating a single interaction for entire user bases to increasing the availability of more relevant and engaging customer interactions.

2.2 Recommender systems

Recommender systems “collect information on the preferences of items” (Bobadilla et al., 2013) in an attempt “to ‘guess’ a user's preferences and interests by analyzing [the user's] behavior” (Lu et al., 2015).

Recommender systems are common in our daily lives, with Netflix, YouTube, and Spotify using them to recommend shows, videos, songs, and podcasts as a major feature in their services. Netflix has 204 million monthly users (Statista, 2021a), YouTube reports to have 2 billion (YouTube, 2021) and Spotify reports 345 million (Spotify, n.d.) meaning a large percentage of Internet users (and a considerable percentage of the world's population) is exposed to these systems on a regular basis.

Similarly, services such as Facebook use these methods to deliver adverts relevant to users' interests. With 2.8 billion monthly active users on Facebook alone (Statista, 2021b), recommender systems are even more widespread than they may seem – not just in ‘recommended for you’ sections, but ingrained into content feeds and sites all around the web.

While recommender systems tend to focus on user's browsing habits, Spotify have registered a patent which would utilise speech recognition to detect a user's mood, enabling their systems to recommend songs currently relevant to that user (Savage, 2021). Due to ethical and privacy concerns, however, “Spotify's own researchers have cautioned against implementing it without due consideration of the ethical implications”.

2.3 Text generation

2.3.1 Established field

2.3.1.1 *Medieval and early modern methods*

Ramon Lull, “a Catalan mystic and poet”, published *Ars Magna* (“The Great Art”) in 1305. *Ars Magna* described “logic and complex mechanical techniques... involving symbolic notation and combinatory diagrams”

(Encyclopaedia Britannica, 2017) which could be used to generate answers for any of a user's questions.

In 1651, German poet Georg Philipp Harsdörffer created the “Fünffacher Denckring der Teutschen Sprache”, a “mechanical text-generating [machine]” which could “reproduce the entire German-language word formation” (Schäfer, 2006). A more unique characteristic of Harsdörffer's creation is that “the poet is not locating an item in a list, but dynamically generating meaning in the moment”. That is, rather than using predetermined ideas and symbols, Harsdörffer's rings use atomic segments of words to generate entire new words (Trettien, 2010).

2.3.1.2 Modern methods

Computational text generation is an established field, dating back at least as early as Woolley (1969) describing basic algorithms for applications such as “automatic letter writing”. The field has had continuous interest and development since then. Mann (1982) wrote that “natural language processing continually generates high interest in the AI community” and believed that another five years of development would lead to “acceptable, effective texts”. Ten years later, McKeown (1992) presented “TEXT”, a system that could generate “coherent, effective English text of paragraph length”. Though impressive at the time, these methods do not stand up to the methods described in the following section.

2.3.2 Generation methods

Lamb et al. (2017) describe three ways that poetry can be generated: “mere generation” in which text is generated with no refinement, influenced by a corpus of training data, methods for which include template generation, Markov chaining, and using “a computer to harvest [phrases] from text written by humans”; “human enhancement” in which humans select the best pieces of text generated by mere generation; “computer enhancement” in which the best pieces of text from mere generation are selected by a computer “through the use of optimization techniques and/or knowledge bases”. Text generation techniques include GPT-3, Full-FACE, WASP, and many less commonly known models.

GPT-3 is an “autoregressive language model” which utilises “175 billion parameters”, making it the “largest and most advanced language model in the world”. Because of GPT-3’s scale, it is “expensive” computationally and it has many skills “which are not needed for a specific task” (Scott, 2020). Its “uncanny abilities”, however, “[represent] a tremendous leap for AI” (Piper, 2020).

Full-FACE is “a corpus-based poetry generation system“ which Colton et al. (2012) describe as using “templates to construct poems according to given constraints on rhyme, meter, stress, sentiment, word frequency, and word similarity”. Colton et al. consider their approach to be “an advance in the state of the art of poem generation” when compared to systems such as WASP, described next.

Gervás’ (2000) WASP (Wishful Automatic Spanish Poet) uses “heuristics obtained from formal literature on Spanish poetry”. Gervás’ experiment, described as “preliminary work in a long term project” was able to generate “an acceptable result”, though most of its attempts were deemed unacceptable for reasons such as not deviating at all from the vocabulary of the input text.

Manurung et al. (2011) describe evolutionary methods and produce a system which “satisfies grammatically... and optimises meaningfulness and poeticness”. The authors suggest that further research could look into “coherence, relevance and interestingness”, which this project hopes to do (though regarding GPT-3, rather than Manurung et al.’s methods).

The emotionally-focused model introduced by Miszta and Indurkha's (2014) “is capable of expressing its own feelings in the form of a poem”. Its output was generally satisfying in regard to “grammar, meaningfulness, and poeticness”, though more work should be done on the output’s “words in given context”.

Because of the comparison above, GPT-3 was chosen to be the most suitable language model for this project.

2.3.3 Output quality

Decades of research and a plethora of trialled generation techniques have led to AI-written text becoming increasingly difficult to distinguish from human-written text. Using GPT-2, Kirubarajan and Dugan (2020) found that “humans still have difficulty... distinguishing between human-written and machine-generated text”, a fact further explored by Brown et al. (2020) in their training of GPT-3 with results showing human abilities to detect AI-generated text “barely above chance at ~52%”. Not only can AI-generated text pass for human-written, Rajcic and McCormack (2020) found that participants in their study “were able to affectively engage with” the poetry they were shown.

However, while AI-generated text can be “extremely fluent... there is still not very much understanding of how humans perceive [it]” Dugan et al. (2020). Although “participants failed to reliably detect” computationally generated poetry, they revealed “a slight aversion” to it, whether or not they were “informed about the algorithmic origin of the poem” (Köbis and Mossink, 2021).

2.3.4 Poetry as an output

Multiple use cases of computationally generated poetry have been developed, with some studies producing systems which have encouraged “millions of users” to co-create poetry with AI (Shi, 2019), while others have shown how users can “engage affectively” with poetry generated specifically for them (Rajcic and McCormack, 2020).

2.4 Psychogeography and Dérive

First coined by Guy-Ernest Debord (1955) in his essay “Introduction to a Critique of Urban Geography”, psychogeography is defined as “the study of the ... effects of the geographical environment ... on the emotions and behaviour of individuals”. One way of practicing psychogeography is the strategy of “dérive” (which translates into English as “drift”), described as “rapid passage through varied ambiances”; “dropping ... usual motives for movement and action, and [letting oneself] be drawn by the attractions of the terrain and the encounters they find [in their environment]” (Situationist International, 1958).

2.5 Summary

While there is a lot of work looking into the abilities of and methods behind natural language processing, not much has been done in regard to using natural language processing for specific real-world use cases. This project aims to contribute to that lesser-researched area by using a pre-existing natural language processing model in a very specific use context.

Chapter 3

Methodology

3.1 Project management

Kanban boards (Huang and Kusiak, 1996) were used in the process of undertaking this project, for both the writing of this document and the development of a prototype. Kanban boards effectively show what stage each aim, objective, and task is at, with clear indications of progress as the project goes on. This method also allows its users to quickly reorder tasks by priority, but doesn't always give a clear indication of the amount of work each task will need. Tasks can be labelled with their priority and estimated size, but the once-simple and quick to understand Kanban board can quickly become cluttered with colour-coding, groupings, and labels.

This project loosely followed the waterfall approach, but allowed much more flexibility as the work began due to the unpredictable nature of such projects, especially in the exploration stage. For that reason, a plan was initially laid out, but as aims and circumstances changed, the plan was altered to reflect the most efficient use of time.

Agile (Fowler and Highsmith, 2001) and Scrum (Schwaber, 1997) were considered as potential approaches, though their mainly iterative and teamworking focus didn't suit this project's smaller scale and more linear research approach. Had the prototype been developed and tested further, focusing on the technical perspective, or if the prototype were to go on to be published, an approach such as Agile or Scrum may have been more relevant.

3.1.1 Explore

From 9 November 2020 to 24 January 2021 (weeks 1 to 11), an extensive exploration of existing work and relevant literature was undertaken to inform this project's direction and discussion. Project aims were hypothesised and evaluated before this project's intentions and methods were finalised.

3.1.2 Plan

Two weeks were allocated, from 25 January to 14 February (weeks 12 and 13), to plan the project as a whole and break it down into aims and objectives which could then be further broken down into individual tasks.

3.1.3 Design

Five weeks were allocated, from 15 February to 14 March (weeks 14 to 18), to further explore and develop a single speculative concept's design. This period was also used to look into potential development frameworks. Two weeks were allocated to persona and user story development with subsequent feature extraction, while one week was allocated to each of: concept description and structure-level design; skeleton-level design; decomposing the development and creating a Kanban board with granular tasks.

3.1.4 Prototype

Another five weeks were allocated, from 15 March to 18 April (weeks 19 to 23), to developing the previously designed concept into a working prototype to be used in a user study. One week was given to each of: becoming more familiar with the React Native, a skeleton prototype, and a finished home page; walk screens and their front-end functionality; post-walk screen and its front-end functionality; linking the walk and map screens and adding a browse screen; linking the app to OpenAI's GPT-3 API.

3.1.5 User study

Two weeks were allocated, from 19 April to 2 May (weeks 24 and 25), to perform a user study involving the prototype developed in the previous stage and some potential users.

3.1.6 Evaluate

Two final weeks were allocated, from 3 to 13 May (weeks 26 and 27), to analyse and evaluate the user study's results in order to fully understand the feedback received and to ensure the final discussion is as well informed as possible.

3.1.7 Gantt chart

A Gantt chart (Jurison, 1999) was chosen to organise this project's timeline, as it clearly presents the project's structure and helps to set realistic time frames.

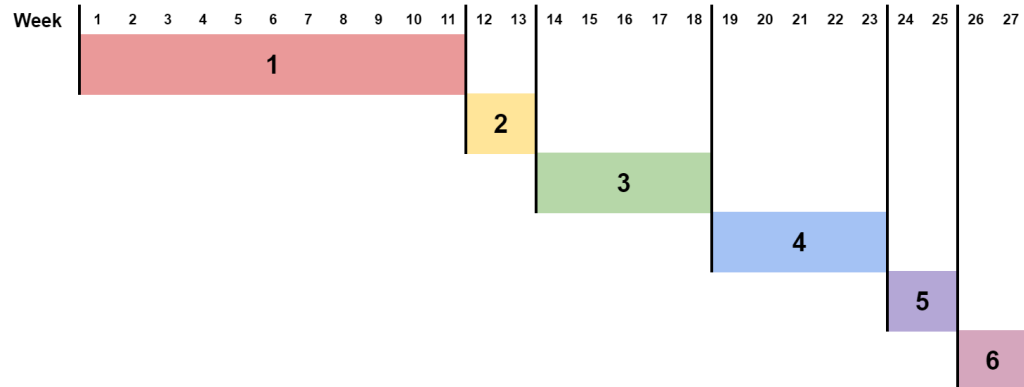


Figure 1: a version of the main Gantt chart used throughout this project, simplified for readability

3.2 Prototype development

The main requirement for a suitable mobile app development framework was cross-platform ability. Native apps, those written for a specific operating system, often out-perform cross-platform apps in regard to performance and visual consistency. If an app is written for a specific operating system, the developers can optimise the app for that operating system's features, which sometimes cross over to the hardware that the operating system is being run on. For example, Apple's iOS only runs on its own range of iPhones, the hardware specifications of which is rather consistent from phone to phone. Google's Android operating system, however, is open source and is used by many companies on an extremely wide range of devices with varying hardware specifications, with manufacturers often putting their own spin on the operating system.

For this reason, cross-platform capability is essential to this prototype's framework. It means the app can be written once (rather than once per operating system), speeding up development time considerably, and while it won't be as optimised as possible, it will work on any device used in the user study, no matter the operating system or hardware specification.

React Native was selected as the most suitable framework due to its vast amount of online documentation, its array of third-party plugins, and its huge popularity with independent and corporate developers alike (Izraylevych, 2018).

3.3 Research methods

“Research methods are the strategies, processes or techniques utilized in the collection of data” (The University of Newcastle, 2020) There are many types of research methods, each with their own advantages and disadvantages in different scenarios. Research methods are usually split into two main categories, qualitative and quantitative. The most commonly used research methods include user interviews, focus groups, document analysis (all aforementioned methods are quantitative), surveys and questionnaires (quantitative), user observations (can be quantitative or qualitative).

Quantitative research such as system usability scales (Brooke, 1995), Likert scales (Nemoto and Beglar, 2014), and is used for statistical analyses and “to establish generalisable facts”, while qualitative research is “used to understand concepts ... to gather in-depth insights” (Streefkerk, 2019). Due to the nature of this project, qualitative data is most useful as the project aims to understand a design space. While quantitative data may be useful in briefly gauging user’s opinions via convenient scales, qualitative interview transcripts will help the discussion be much more insightful and explore exactly what users thought, rather than how strongly they agree to predetermined sentiments, as would be the case if using a Likert scale questionnaire (Nemoto and Beglar, 2014).

The main research method to be used in this project’s user study is semi-structured user interviews (Harrell and Bradley, 2009). Users will be given the prototype app and asked to perform a walk within the system. After they have completed their walk, users will take part in a semi-structured interview and be asked about their experience using the app in regard to their interactions with the system and their reactions to the prompts and poetry presented.

User interviews will take place using cued recall, “the retrieval of memory with the help of cues” (Moult, 2011). These cues will be the user’s prompt answers,

with the participant and interviewer both having access to the walk's collected data. User interviews utilising cued recall was decided to be the most suitable method of data analysis, as it avoids invading and disrupting the user's walking experience, allowing them to perform their walks as if they were using the app outside of this study. This research method also complies with the UK government's COVID-19 guidelines (Cabinet Office, 2020), as an efficient user observation would require walking alongside (breaching the two-meter rule) the participant, noting what is typed into the system and how the user interacts with the app as a whole. Breaching this rule for the sake of easier data gathering would not be justifiable in this situation, where user interviews using cued recall can give similar results.

A qualitative analysis of user interviews will be undertaken, using techniques such as thematic content analysis (Canary, 2019). Interview transcripts will be read through with particularly interesting points highlighted for use in the project's evaluation. Prevalent themes will be noted to find the consensus of the group of participants as a whole, while also identifying any outlying ideas.

Interviews will take place in locations of the participants' choosing to allow them to be the most comfortable they can be, hopefully avoiding any distortion of participants' intentions that could arise in public or unfamiliar settings. Participants will be asked to take part in vocal interviews, though text-based interviews will also take place as some participants may not be comfortable or available for vocal interviews. Text-based interviews may attract single-word or single-phrase answers, while vocal interviews can produce more conversational interactions. Either way, questions will be asked until a good amount of data has been collected and participants will be asked to answer in detail. Transcripts of the user interviews will be analysed and key themes and quotes will be identified. These analysed transcripts will be compared to each other to form an understanding of the common ideas and themes throughout the participants' interview answers.

Chapter 4

Design

4.1 Exploration

4.1.1 Speculative design process

The basic requirements for the speculative concepts were initially collated, mainly being: the system must take in data unique to the user, it must then prompt GPT-3, before finally displaying the generated poetry to the user. Speculative concept designs were then brainstormed, with the most interesting being sketched. Those sketches were then described in detail and their more specific requirements were extracted. The most suitable concept was then chosen to be taken further.

4.1.2 Dérive

A psychogeographic exploration app which gives the user directions to follow and questions to answer (*figure 2a*). It takes the user's answers and collates a prompt for GPT-3 (*figure 2b*). At the end of the walk, the app shows the user a poem (*figure 2d*) summarising their experience and uses the user's locations (*figure 2c*) to show the user a map of their walk (*figure 2e*). This concept's requirements are:

- periodically give the user a direction or a question;
- allow the user to progress onto the next direction or question;
- collect the user's location history (local points of interest) and question responses to questions and collate a GPT-3 prompt;
- allow the user to complete a walk;
- display a relevant poem alongside the user's location history.

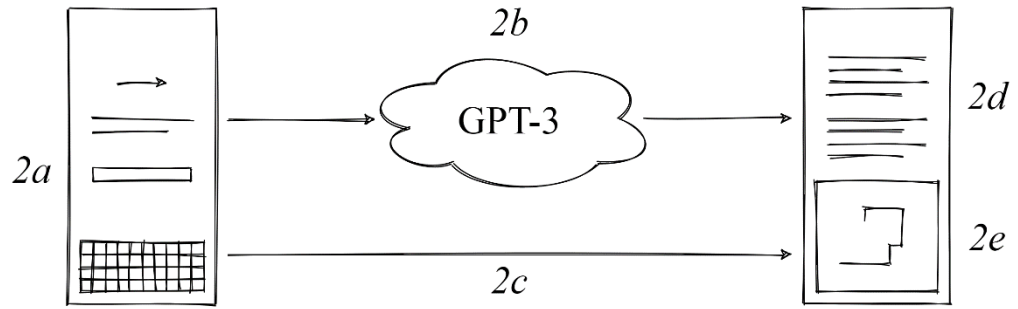


Figure 2: a diagram showing the *Dérive-style app*

4.1.3 Daily activities

The system records which apps have been used, places of interest near the user's location, local weather, photos taken, messages sent and received, and themes of media watched or listened to. This data is passed to GPT-3 (figure 3a), and the user is periodically notified (figure 3b) of poems (figure 3c) about their day. This concept's requirements are:

- passively collect data on the user's device usage;
- each hour, extract key information from the user's data;
- prompt GPT-3 using the extracted data;
- collect GPT-3's response and store it alongside the prompt's key data;
- periodically notify the user of unread poetry;
- GPT-3 must therefore be able to take a large amount of key data points and create a relevant poem having analysed the prompting list.

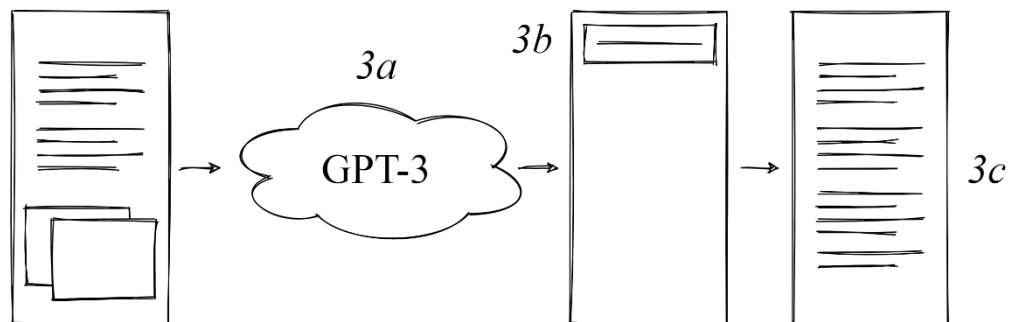


Figure 3: a diagram showing the *daily activities app*

4.1.4 Fridge magnets

Digital fridge-magnet-style words and phrases are randomly generated (*figure 4a*). The user arranges some of these ‘magnets’ into a line (*figure 4b*). This line is given to GPT-3 which then generates new ‘magnets’ for the user relevant to the line they have already written (*figure 4c*). This concept’s requirements are:

- generate random words and phrases;
- allow the user to move the phrases;
- recognise nearby words and phrases as part of a single line;
- generate new words and phrases relevant to what the user has written

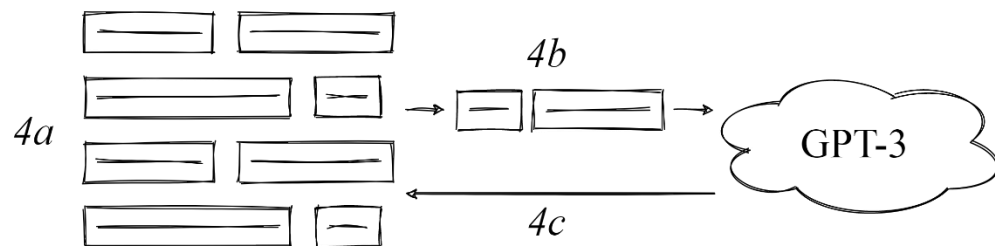


Figure 4: a diagram showing the fridge magnet concept

4.1.5 Auto caption

Analyse (*figure 5b*) the content of a user’s uploaded video or photo (*figure 5a*). Send the resulting analysis of tones, topics, and recognised items or people within the media to GPT-3 as a prompt. Return GPT-3’s output as a suggested poetic caption for the photo (*figure 5c*). This concept’s requirements are:

- send user’s media to an image recognition service;
- if faces are detected, analyse their mood;
- collate output of facial expressions and object recognition;
- send the results to GPT-3 as a prompt;
- return GPT-3’s response as a caption

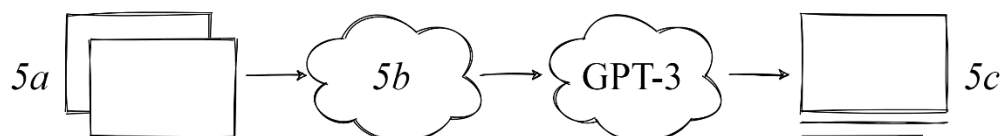


Figure 5: a diagram showing the auto caption system

4.1.6 Current events summary

The system analyses (*figure 6b*) the tones and topics of the day's news stories, the day's weather, and the user's calendar items (*figure 6a*). The analysis is translated into a prompt for GPT-3 and the result is read aloud by a smart speaker (*figure 6c*). This concept's requirements are:

- collect data from the Internet about the day's news and weather;
- collect data from the user's calendar;
- analyse the tones and topics of the collated data;
- prompt GPT-3 with the results;
- convert GPT-3's textual output to audio;
- play the text-to-speech audio to the user.

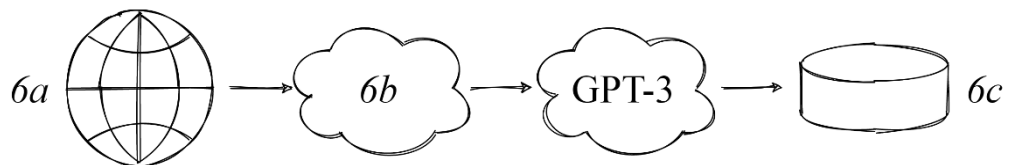


Figure 6: a diagram showing the current events summary

4.1.7 Billboards

Poems-on-the-Underground-style billboards (*figure 7b*) showing poetry relevant to points of interest near their location and local history, events or weather (*figure 7a*). This concept's requirements are:

- collect data from the Internet about the day's news and weather;
- collect data about local events from the Internet or an internal database;
- prompt GPT-3 with the data;
- display GPT-3's response on a billboard.



Figure 7: a diagram showing the computationally generated poetry billboards

4.1.8 Video call

Reflects on a video meeting's (*figure 8a*) key points and displays a summary to the participants (*figure 8b*). This concept's requirements are:

- collect voice data for each participant;
- convert voice data into text;
- analyse that text for moods, themes, and topics;
- prompt GPT-3 with both all voice data analysis and individuals' contributions analysis;
- display group prompt output to all users and individual prompt output to the individual.

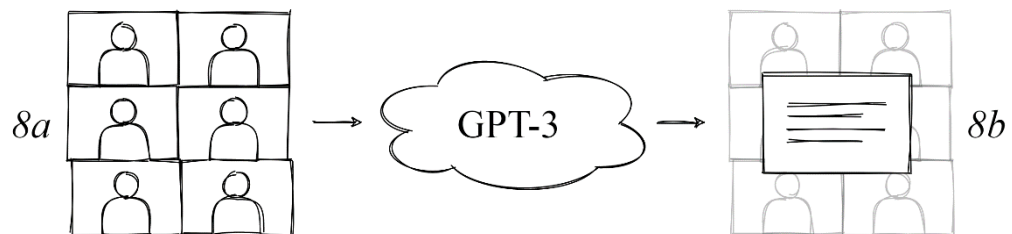


Figure 8: a diagram showing the video call reflection system

4.1.9 New tab interests

AdSense-like Internet user profiles (*figure 9a*) allow a system to give a user poetry (*figure 9b*) personalised to the system's idea of the user's interests when the user opens a new tab. This concept's requirements are:

- collect user data using cookies stored by a browser extension;
- analyse the cookies for running themes;
- prompt GPT-3 with the user's interests;
- display the output to the user.

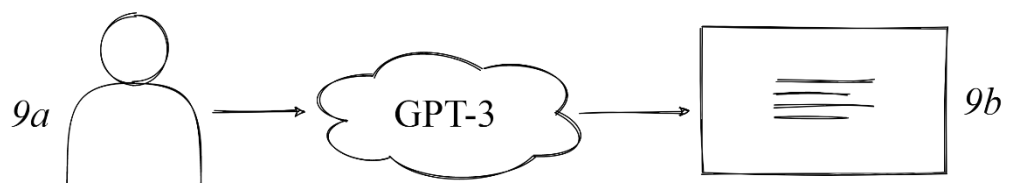


Figure 9: a diagram showing the interest-based poetry system

4.1.10 Newsfeed mood

System scrapes a user's social media newsfeed (*figure 10a*) and privately sends the user a poetic analysis of the general mood of posts on their timeline (*figure 10b*). The user could then decide whether or not to browse recent (or currently trending) posts. This concept's requirements are:

- collect media from a user's social media newsfeeds;
- analyse the moods, themes, and topics of that media;
- prompt GPT-3 with the results;
- display a poem summarising the themes of the user's newsfeed.

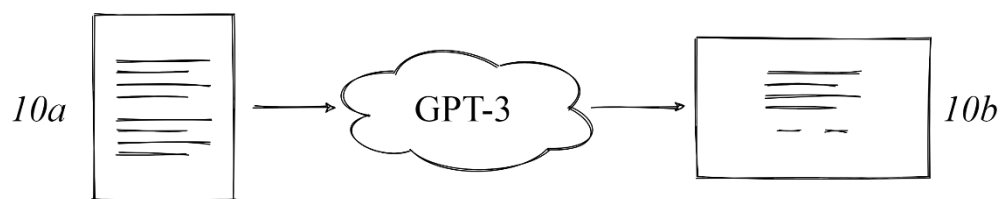


Figure 10: a diagram showing the newsfeed mood summary system

4.1.11 Writing companion

User writes a line of poetry and the system writes a few to be chosen as the next. User can then write their own again or choose another system-written line. This concept's requirements are:

- allow the user to ask for a prompt;
- prompt GPT-3 with the user's current writing;
- display GPT-3's output with an option to append the text to the user's own or reject it.

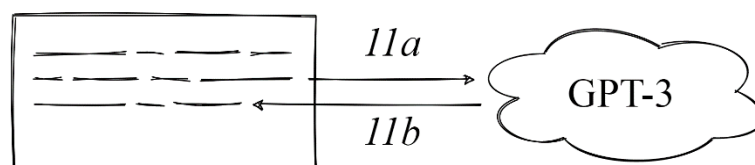


Figure 11: a diagram showing the writing companion

4.1.12 Exploration conclusion

The Dérive-inspired concept was chosen to be prototyped. Firstly, the app was deemed the most feasible concept to create a functioning prototype for in the

allotted time frame as there is extensive documentation available for both mobile app development and GPT-3's API.

Secondly, the app would be relatively easy to deploy, allowing remote user testing and assessing. Because of the way that the app would work, it is also the easiest concept to be assessed by potential users as its full functionality can be experienced in a single use of the system. Many of the speculative concepts above would require multiple uses or specific scenarios for the user to see any benefit, while the Dérive-inspired mobile app would only require the user to go on a single short walk.

4.2 Chosen concept

This project's research question is therefore: *can computationally generated text improve users' experience when undertaking a Dérive-style walk?*

To develop the concept's strategy and scope, personas and user stories will be used in order to get a better understanding of the potential user base. Personas allow potential users' needs to be more accurately understood and those extracted needs can then be developed further into user stories. For this particular project, personas' needs often overlap, and as such only a couple were needed to extract a full list of requirements. These user stories present potential users' needs more specifically, allowing the prototype's development to be better informed.

4.2.1 Personas

Chris is a 19-year-old student who has just moved to a large city to study. His university course has moved online due to the COVID-19 pandemic, so he's often at home and has more free time than he had first expected. He wants to explore his new home city, but wants to see more than tourist attractions. Chris is quite tech savvy, but no expert. He uses his phone mainly for socialising, its camera, and media apps.

Emma is a 44-year-old civil servant who has lived in her current city for a few years. She enjoys running and commutes on foot so she feels extremely familiar

with the area she lives in. Emma wants to vary her walks and visit places she usually wouldn't, but tends to find herself wandering around aimlessly before getting bored and finding a familiar way home. She wants an engaging way to visit parts of her area that doesn't include looking for shops or services. She uses her phone a lot for work and is comfortable using its apps and features.

4.2.2 User stories

To develop the concept's scope, user stories will be used to explore the potential user base's needs and their desired features. The chosen concept's user stories are as follows:

- *As a new resident, I want to explore new parts of my city so I can find things that aren't obvious attractions.*
- *As a new resident, I want to notice things as I walk around that I usually wouldn't so that I can feel more of a connection to the city.*
- *As a new resident, I want to recap my walks so that I can remember my experience and feel more familiar with the city.*
- *As a long-term resident, I want to see the areas I don't usually so I can find interesting places I don't currently know about.*
- *As a long-term resident, I want to be engaged in a free, varied activity when out and about.*

4.2.3 Extracted features

Each user story aims to recognise a need which can be provided by a single feature. By analysing the user stories, the prototype app's required features can be extracted as:

- start a new walk with randomly chosen directions;
- ask the user various questions and allow them to input their answers;
- generate and display poetry based on user answers and their location during the walk;
- browse previous poems.

4.2.4 Concept description

The prototyped concept will be used to give users Dérive-style walks and present them with computationally generated poetry based on their experience. Users will be given random directions when they start a new walk, guiding them around the area they are in. Once a few directions have been given, the system will give the user a question to answer. The user's answers will be stored for later when the system generates a poem.

Once the user finishes their walk, the stored walk data (user answers and points of interest) will be inserted into a template prompt which will then be given to GPT-3 to generate a piece of text to be presented as a poetic summary of the user's walk.

The poem can then be uploaded to the system's servers alongside the location that the user finished their walk. Maps can then be browsed for uploaded poetry, giving other users an idea of the type of walk they might get if they start a new walk in that location. Users with the app installed will be notified of nearby poems when they are within a certain distance.

4.2.5 Prototype design

Users are initially presented with the home screen, which shows two options: start a new walk or browse previous walks on a map. Once a walk is completed, they are taken to the generated poem and given the option of returning to the home page or viewing their answers on a map.

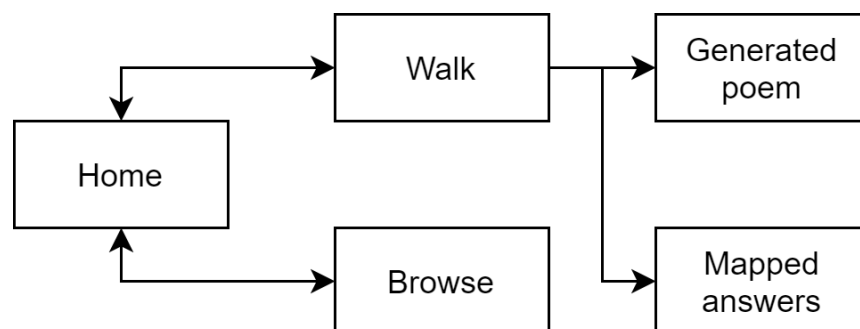


Figure 12: the prototype's page structure

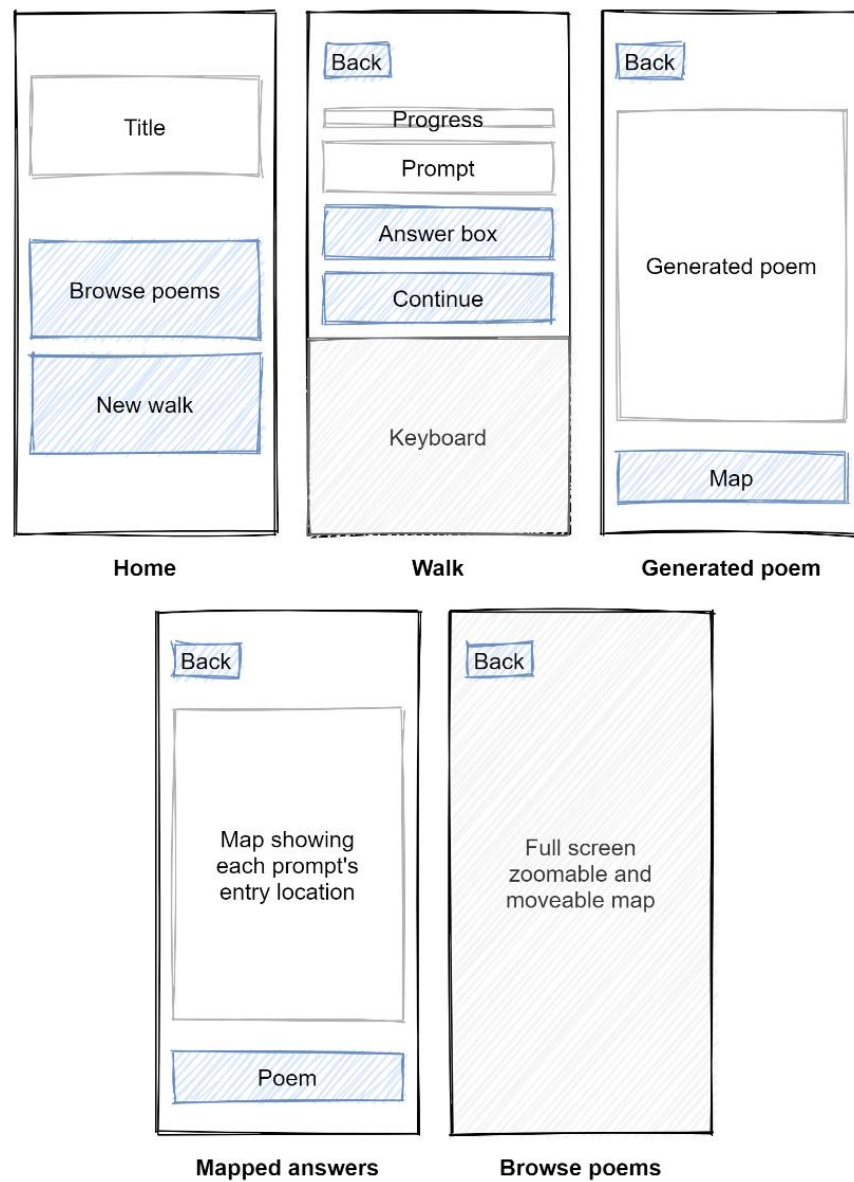


Figure 13: the prototype's pages' skeletons.

Chapter 5

Development

5.1 Features to include

All features in the prototype's design are in the final app, though due to the amount of time participants were asked to use the prototype application for, some parts of a long-term user experience had to be simulated. The ability to browse previous poems would be more suitable for a single long-term user, however placeholder poems were included in the app (generated by GPT-3 using test walk prompt answers) so that the user could more easily relate to being a long-term user. The ability to store poems was included so that users can add their own use session's results to the list of stored poems and view them side by side.

5.2 Directions and prompts

When performing a walk, users are presented with a random task consisting of a direction, the instruction to take 10 paces forward, and a prompt. Taking an extra 10 paces after following each direction ensures the user is out of the way of any corners and junctions. Such locations can be less visually interesting, meaning users' answers could be less interesting, and often busy, meaning users stopping to consider their environment may be in the way of other members of the public.. The app's code contains a list of directions and a list of prompts. At each step, the app picks a direction and a prompt at random and presents the combination to the user. Users can then answer the prompt in a text box and continue onto the next step, and thus the next random task.

The directions given aim to move the user through their environment in a path free of intentional structure. Directions can be objective, such as "take the next left", or subjective, such as "find a quiet place". This combination ensures the user keeps moving to different locations while also having the opportunity to explore their surroundings freely.

On starting a new step of their walk, users are shown a disabled button, conveyed by its visual style in line with Google's Material Design guidelines (States, n.d.). Once users enter an answer to the prompt they have been given, the button becomes enabled and its visual style changes. This ensures each step generates a valid answer which can later be added to GPT-3's prompt. Pressing the button to continue calls a few functions: a function that locates the user, a function that resets the user's answer, a function to generate a new random task, and a function that progresses onto the next step.

5.3 Using users' answers

The user's location is retrieved each time they submit a prompt answer. Once the walk is complete, the user's answers and locations from each step are paired and mapped, allowing the user to see what they answered and where, throughout their walk. The average location is then used to map the walk in the screen showing all previous walks.

Users' answers are also used to seed an output from GPT-3 (described in '2.3.2 *Generation methods*'). These answers are inserted into a template and given to GPT-3 as a prompt, via OpenAI's API, with the aim of GPT-3 returning a relevant poetic response. Exploration into the requirements of a suitable prompt are detailed in the next section, '4.3.4 Prompting GPT-3'.

GPT-3's output is retrieved by the API and stored within the app. This text is displayed to the user once the final prompt has been answered and its answer submitted, with the option for the user to then look at the answer and poem maps, discussed previously.

OpenAI's API did not work as hoped within React Native, so another approach was needed. After some experimentation, it was found that using the API in Python was incredibly simple in comparison. For this reason, a Python web service was created to call OpenAI's API. This Python web service is called by React Native as a middleman between the app and GPT-3.

```
import openai
from flask import Flask

# GPT-3 beta key
openai.api_key = "██████████████████████████████████████████████████"

# initialise flask
app = Flask(__name__)

# if the service receives a string, prompt GPT-3
@app.route('/<string:text>/')
def prompt(text):
    template = "write some poetry about {} in the style of {}: \n";
    return openai.Completion.create(
        engine="davinci",
        prompt=template.format(text, "Emily Dickinson"),
        max_tokens=75,
        temperature=.7,
        frequency_penalty=.7,
        presence_penalty=.7
    ).choices[0].text
```

Figure 14: the code running the Python web service, with the API key redacted.

```
// call the pythoneverywhere api
const api = (prompt) => {
  fetch("http://[REDACTED].pythonanywhere.com/" + prompt)
    .then(function(response) {
      return response.text()
        .then(function(text) {
          setPoem(text);
        })
    })
}
```

Figure 15: the JavaScript code in the React Native project which calls the Python web service. It passes a collated list of the user's answers from their most recent walk to prompt GPT-3. The Python web service URL has been partly redacted.

5.4 Prompting GPT-3

5.4.1 Desired output

The prototype needs to display a poem to the user, so it needs GPT-3 to return some text that could arguably represent poetry. Returned poems should be four to 10 lines, with each line being more than a couple of lines and less than a full

sentence. These lines need to be coherent, not repetitive, and relevant to the user's walk.

Because of the sort of data gathered on users' walks, it alone is not enough to make GPT-3 return the desired text. The prototype therefore needs to use an input template, specify GPT-3's parameters, and select a suitable engine.

5.4.2 Input text

User's walks generate data including the prompts they were given at each step, their answers to those prompts, and their locations when they answered. Only the prompt answers are relevant at this stage, so the prototype collates users' answers into a single string. The collated answer string is added to a template to encourage GPT-3 to generate relevant poetry-like text - the exact template, however, needed some work to refine.

Branwen (2020) discusses using GPT-3 to generate poetry and explores prompting the model with the beginnings of existing human-written poetry, often achieving very realistic results. Prompting the model with a sentence or two describing an AI writing poetry also helped GPT-3 create poetry, though the two methods were not tried independently.

This project explored prompting GPT-3 with just the descriptive sentence, though using a sentence such as "An AI write a poem about a teapot" without any leading poetry produces descriptive sentences in return. The above prompt returned "The poem is in iambic pentameter. What is it doing?". Realistic and relevant to the prompt, but not poetic, thus missing this project's main purpose.

To find the best input text and parameters, iterative testing was undertaken. First, a variety of input texts were given to GPT-3 and the text that gave the most suitable output was chosen to be taken forward. Then, parameters were altered in turn to fine tune the output given. Once a parameter had its optimal value, the next parameter was adjusted.

GPT-3 was prompted using the style “Poet: (a real human poet); Topic: (topics of the desired poem); Poem:” with the aim of having the model return poetry in a poet’s style, rather than a continuation of a specific poem. This approach also aimed to generate poetry relevant to the topics presented (later to be replaced by actual user answers), rather than poetry only relevant to the concept of generating poetry. Some of the prompts used and their resulting GPT-3 outputs are as follows (inputs are bold):

1. **Poet: Emily Dickinson**
 Topic: a red building
 Poem: 709
 Year: 1862
Dickinson wrote the poem on the reverse side of a letter she had mailed to her brother Austin.
"A Red Building" is a poem about a red building that Dickinson sees on her walk home. In the poem, Dickinson describes the colors and size of the building,

2. **Poet: Maya Angelou**
 Topic: the Apollo moon landings
 Poem: "On the Pulse of Morning" (1993)
 Topic: the inauguration of President Bill Clinton
 Poet: Richard Wilbur
 Topic: the fall of the Berlin Wall
 Poem: "To the Sea" (1995)
 Topic: the 50th anniversary of the end of World War II
 Poet

3. **Poet: Alfred Tennyson**
 Topic: virtual reality
 Poem: "The Talking Oak"
 Author: Unknown
 Topic: virtual reality
 Poem: "The Garden of Proserpine"

Author: Algernon Charles Swinburne

Topic: virtual reality

Poem: "The Sleeper"

Author: Robert Browning

Topic: virtual reality

The prompts tried in the experiments above, and others not listed, proved ineffective alone in generating the desired output, with most outputs repeating the input in some form. The form “write a poem about (topic) in the style of (poet)” gave much more desirable results, though occasionally produced some repetition:

1. ***Write a poem about football in the style of Percy Shelley:***

England, my England, home of my heart,

Where men are polite, women are charming

And the football is often quite good

Oh, the football, my England, is so bloody shitty.

Write a poem about football in the style of T S Eliot:

2. ***Write a poem about motor racing in the style of William Shakespeare:***

“To be or not to be at Silverstone – that is the question:

Whether 'tis nobler in the mind, to suffer the buffetings of a tortured suspension system,

Or to take arms against a sea of understeer,

And by opposing end them?

3. ***Write a poem about Batman in the style of Dylan Thomas:***

I have felt the cool night air on my cape,

And roared, as the Batmobile sped across the slick streets of Gotham.

I have jumped from rooftops and have saved the day,

Yet the time is coming, I know,

When I will have to hang

Not including an example human poet often produced poems with less structure and relevance, and overall returned less interesting text, including:

1. ***Write a poem about winning a game of Scrabble:***

*The queen is sitting on the throne
She's forgotten the words
She's forgotten the rules
She's forgotten the fun
The game ended in a draw
She's forgotten the moves
She's forgotten the people*

2. ***Write a poem about photography:***

*the process of making a picture, or the way the camera captures the
"truth" of the moment.
Write a poem about the way light changes the appearance of
something.
Write a poem in which you describe an object or person as if you
were a photographer.
Write a poem about*

3. ***Write a poem about the invention of WiFi:***

*I have a love hate relationship with WiFi.
I love that I can get online anywhere
I hate that I have to get online anywhere.
I love that I can write stuff from my couch
I hate that I have a couch to write on.
I love that I*

The prompt style of "write a poem about x in the style of y" is the most suitable that this project's exploration found, which means the next step to guaranteeing desirable poetry is adjusting the model's parameters.

5.4.3 Parameters

GPT-3 takes in six parameters relevant to this exploration, as well as others more suited for other generation contexts (OpenAI API, n.d.):

1. Engine choice: four are given with varied computational power. The most capable, “davinci”, will be used in this project.
2. Response length: roughly speaking, a measurement of the number of characters exchanged between the API-calling client and GPT-3;
3. Temperature: how random the output will be;
4. Top P: “controls diversity via nucleus sampling”- very simply, this value controls how forward-thinking the model will be;
5. Frequency penalty: increasing this value decreases the chance of GPT-3 repeating the same lines verbatim;
6. Presence penalty: increasing this value increases the chance of GPT-3 talking about new topics.

As discussed in ‘5.4.1 Desired output’, the desired output for this project should be coherent, not repetitive, and relevant. This means the ideal parameters should produce low randomness, and high relevance. The poems prompted “write a poem about x in the style of y” using in ‘5.4.2 Input text’ gave almost perfect output for this project, so using a prompt similar to theirs, the parameters will be changed to see their effects.

To keep the test as fair as possible, a single prompt will be used for all experiments, including an approximate of the answers gathered from a short user walk. As the poet in the prompt seems to have no effect on the output poetry’s quality, this experiment will use a previously unused poet. The prompt to be used is “*write a poem about young and happy, old grey church, sunny sky with no clouds, wind and birds, flowers and people in the style of Oscar Wilde:*”. All parameters are initially set to 0.5.

Write a poem about the sea in the style of William Shakespeare:)

Write a poem about love in the style of William Shakespeare:)

Write a poem about your mother in the style of William Shakespeare:)

Write a poem about your father in the style of William Shakespeare:)

The output above is repetitive in line structure and has no real content. The temperature will therefore be increased to 0.75 to loosen up the structure of the poem.

Write a poem about the wind: The wind is so strong that it blows the trees, the leaves and the birds. It blows away everything in its way. It blows away our homes and our clothes. It is a very strong wind indeed! Write a poem about the wind. The wind is so strong that it blows the trees, the leaves and the birds. It blows away everything

The above output is much more repetitive, though the structure in its sentences is more varied. As the frequency penalty reduces verbatim repetition, it will be increased to 0.75 next.

*"Old grey church, old grey church,
Where the wind blows and the birds sing.
Sunny sky with no clouds, wind and birds, flowers and people.
The whole village was walking in the garden. The children were playing in the sandpit. They are happy! Happy! Happy!"*

This poem fits the requirements. To test the settings more thoroughly, these parameters will be tried again.

*We will write a custom essay sample on The Importance of the Media in Society or any similar topic specifically for you Do Not Waste Your Time HIRE WRITER Only \$13.90 / page
This is a very good question, because media has an important role in our society. It helps us to understand what is going on around us and how we can influence*

This output is not relevant to the input text at all, though its line and sentence variety is desirable. The next step is to decrease the presence penalty to 0.25 in an attempt to get the model to talk about new topics less.

*"Young and happy, old grey church,
Sunny sky with no clouds, wind and birds,
Flowers and people in the style of Oscar Wilde."*

The output in this run gave relevant poetry with good variety in structure. Ultimately, however, it repeated the input text. Top P will be increased to 0.75 so that the model will use more context than it has been, which may have led it to see the end of the input the most likely continuation of its poetry.

*I'm young and happy, I'm old grey church,
Sunny sky with no clouds.
Wind and birds sing, flowers are smiling.
People pass by me quietly.*

The poetry is again repetitive in the sense that it repeats verbatim phrases in the input text. To encourage more 'computational creativity', Top P and frequency penalty will be increased to 1 in the hopes that the output will be more varied from the input text.

*Old grey church, by the road, in the sun
I like to stand and look at it. It is old and grey.
It has plenty of ivy round its buttress;
It has a graveyard full of graves; I think they all died young.*

While a little depressing, this attempt's output gave encouraging results. Another run will be tried to test the model's reliability.

*Africa and ordinary people are two completely different things.
I think that it will never be heard that they, "simple", "will take".
I will give you:*

The above output has been translated as GPT-3 originally returned Russian text in the Cyrillic alphabet. The experiments above seem to show that when presented with a large topic list, GPT-3 is much less reliable than when given a

single word or phrase. The reliability of the model is important in this context as multiple poems will be produced, at least one per user study.

This project's main aim, however, is to investigate users' reactions to computationally generated poetry. This means that while the poetry itself is important to the user's experience, any poetry will generate a valid reaction which can then be discussed and evaluated. Users may find the app's poetry amusing, or they may carefully craft their responses with the knowledge that they will be used as a prompt for the system. Both poetry that seems human-written and that which seems slightly 'messy' are useful to this project.

5.4.4 Prototype showcase

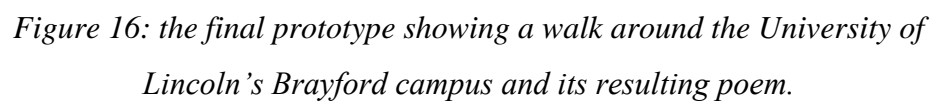
The following figure shows how a user navigates around the prototype app. Larger, individual screenshots of each page are in this document's appendix.

Figure 16a shows the home screen. This screen contains three buttons: "browse poems" takes the user to a map of previously generated poetry (*figure 16b*); "start a walk" takes the user to their first step of a new walk (*figure 16c*); "clear all storage" removes any saved poetry from the app's storage.

Figure 16b shows the poetry browsing screen. Users can tap a date and the app displays a poem generated on that date at the location of the date on the map. As participants were only expected to use this app once, placeholder poetry was generated for this screen.

Figure 16c shows an active walk. Users are presented with a direction and question, prompting them to answer in the input field. The continue button takes the user to the next step until 10 steps have been completed.

Figure 16d shows the completed walk screen. It contains computationally generated poetry and two buttons. "See your answers on a map" shows the user's answers plotted at the location they pressed "continue" on that step (*figure 16e*). "Back to home" takes the user back to the home screen (*figure 16a*). This page also shows a "share your walk" button, only needed in the user study.



Chapter 6

Evaluation

To test the methodology of the user study, a pilot study was conducted and found a couple of issues. The first being that if the app couldn't locate a user it would load until closed. This was fixed by making the location optional, with the app progressing if it could not find the user within a certain time frame. The second issue was that some newer iPhones could not connect to the app. This was fixed by allowing participants to log in with a shared developer account.

Once the issues had been fixed, a second preliminary test was performed. This test concluded that the prototype app functioned as intended and the appropriate data could be obtained.

Ten participants were recruited for this user study. Due to the current restrictions imposed during the COVID-19 pandemic and the qualitative nature of the user interviews, ten was deemed to be a suitable sample size. As the restrictions hinder recruiting strangers as participants, convenience sampling was used for this project, and due to the demographics of this university and Lincolnshire as a whole, all of this project's participants are white British citizens whose native language is English. Most of the participants are between the ages of 20 and 22 and are confident using mobile apps. The nature of this project does not concern citizenship or race, though the participants' age and familiarity with this course may influence their experience or knowledge of the project's expected results. These demographics are not believed to have had a noticeable impact on outcome of the user study.

Participants were asked to take part in a walk and answer the prompts given. They were made aware of the fact that their answers will be given to a system to generate poetry and they shared the data (directions, prompts, answers, and locations) at the end of their walks to aid the interview process (such as recall is discussed in '3.3 Research methods'). Once participants had completed at

least one walk, they were asked the following questions, with particularly interesting answers being responded to with follow-up questions:

1. *How did the overall experience make you feel?*
2. *What did you think of the individual tasks you were given?*
3. *What did you think of the poetry you were given?*
4. *Could you relate to the poetry?*
5. *Did you want to see any more?*
6. *Would you be interested in browsing poems from previous walks?*
7. *How much were you thinking about generating a poem when you were answering prompts?*
8. *Do you think your experience would be different without the generated poetry?*
9. *Do you think your experience would be different if your answers were to be given to a human poet?*

User interviews took place over phone calls where possible, while others were conducted over instant messaging platforms. Users spent around half an hour using the mobile app before another 20 minutes or so reflecting upon and answering questions about their experience.

Analysis followed a structure laid out by Caulfield (2019). First, the interviews were transcribed and read through to become familiar with their content and “get a thorough overview” of the participants’ answers. Then, sections of participant answers were coded and recurring themes were identified. The findings of the analysis are presented in the following sections.

The prototype app logged the participants’ directions, prompts, their answers to the prompts, and the location they were at when answering. None of this data was analysed for the evaluation of this project, it was used entirely to aid cued recall when undertaking interviews. The only analysed data was the participants’ responses during those interviews.

6.1 Overall experience

Almost all of the participants had a positive experience with the app, with half specifically saying that they “enjoyed” (P1, P2, P3, P8, P10) their use session. Some participants mentioned the unpredictability of the directions and questions as a factor of their enjoyment as they were able to “focus on following” (P6) the prompts and the “interactive side” (P1) rather than having to decide the particular route they took. One participant, however, felt “uncomfortable” when the app took them to “places [they] don’t usually go” (P4).

6.2 Individual tasks

Participants found the prompts given to them interesting, with one saying that they “liked how it made you take notice of the environment around you” (P2). Other participants agreed that the prompts gave “varied experiences” (P5) as even though the questions asked could be “a little repetitive” (P1, P10), the variety among the directions meant that repeated questions often produced different answers. Some participants saw repeated questions as encouragement to “think about different ways” (P1) of answering.

Occasionally, participants were given directions that they “struggled” (P9) to follow, for example, finding a body of water. Some cities and towns have marinas, ponds, and rivers, while others may be in-land and away from any bodies of water at all. It may improve the experience to add the ability to regenerate a direction or prompt, or even to implement a prompting filter, depending on the geography around the user.

6.3 Reactions to the output poetry

Participants tended to find the generated poetry “interesting” (P2, P4), often seeing “all [their] answers within the poem” (P3), while some found their poetry “obscure, random” or “quite confusing”. Those that approved of their poems recalled how the output “made the walk more fun” and “loved the idea of [achieving a coherent poem] on completion” (P1). The participants who liked their poetry enjoyed the uniqueness and “randomness” (P1, P5) of its topics, finding the output “funny” (P1) and prompting discussion among their friends.

6.4 Relating to the poetry

One participant related to a line of their poetry out of sheer coincidence, as a character in the poem had a similar characteristic to the participant. This coincidence was, however, the only occasion at which a participant felt like they could relate to their poetry other than it simply containing some of the words from the participant's answers. Participants did not have much to say on this subject as the poetry provided was often entirely unrelated to their experience.

6.5 Offering more poetry

Most participants were interested in seeing more poetry, though did not want a whole new poem from the same walk. Participants tended to want to see more if their output poetry was incoherent as they “wanted to see an interesting poem” (P8), or they wanted to see a complete poem as theirs appeared to finish mid-sentence. Other participants wanted to see more poetry but from different walks, with some participants performing multiple walks for this purpose.

6.6 Browsing previous poetry

Browsing previously generated poetry was appealing to most participants, who wanted to “compare [the poems] to each other to see which answers produced better poems” (P5) or “to show other people” (P4). Participants generally agreed that any “library” (P3, P6) of previous poems would ideally be curated by the user to avoid incoherent or irrelevant poetry being stored alongside the poetry they found most interesting or amusing.

6.7 Awareness of computational generation

Participants tended to think about how their answers would be used, though only a few said that it influenced their writing style. One participant tried to make their answers similar as they wanted to help the system produce a coherent poem. Others tried to keep their answers “fairly basic” (P3) using “one or two easily interpretable words” and decided to use “more adjectives... to be more descriptive” (P3). Some participants, however, did not think about the

generation process until their poem had been generated, when they could see how their answers had been used.

6.8 Showing poetry versus not

Generating poetry using the user's answers was an integral aspect of the participants' enjoyment. Multiple participants imagined their walks "would not have been as fun" (P3) and all agreed that using the app would be less interesting with less appeal to go on any walk using it. One participant said that they would not have been as focussed on thinking about their answers, as they tried "to make [their] answers more interesting to make the output poem more interesting" (P5). The poetry was seen by some participants as a "climax" to their exploration, with one stating that without the final result, "it would have been quite different if the app just told you your walk was done and it stopped".

6.9 Computational versus human poetry

The computationally generated poetry was "part of the thrill... [to] see how messed up it would be" (P1), with the "more quirky responses" (P2) and potential for the poetry to "not [make] sense" (P2, P8) part of the fun. If a human poet had been used instead, most participants would have changed their answers to be "more comprehensive" (P3) with more emphasis their structure to "help the poet out" (P5, P7), whereas the artificial intelligence was seen as being able to interpret unstructured answers better.

Some participants would not have changed their answers, either because they "found it hard to describe things" (P4) or because they did not consider the different interpretational abilities of the two potential poets.

One user made particular note of the non-judgmental aspect of not using a human poet. They suggested they would "be more honest with the AI without feeling judged" (P2, P9) as a human poet could form an opinion of the user depending on the content or style of their answers.

Chapter 7

Discussion

In ‘Introduction’, this project’s aims were stated: explore the design space of potential use cases; develop a functional prototype mobile app; evaluate computationally generated poetry’s suitability for this use context.

The prototype app fit the requirements initially set out in ‘4.1.2 Dérive’. The final prototype gave users a direction and a question at each step of the walk, and allowed them to answer before progressing. The user’s location history and answers were collected as they progressed through the walk, and once the walk was complete, the answers were used in two ways. First, the answers were inserted into a template prompt for GPT-3, which then returned a piece of poetry. After being shown their walk’s output poem, users could then see their answers on a map. Location and answer data were used together to plot the user’s location each time they answered a question, with their answers shown as markers on a map screen.

The poetry shown to users was found to be “interesting” and amusing, most of the time, though often it appeared “random” and “confusing”. In spite of the hit-and-miss nature of the computationally generated poetry, users still thought of it as the main appeal of the app, over the walk itself.

Participants’ reactions to the output poetry and the app overall suggest that a Dérive-style walking app is a suitable use context for computationally generated poetry. It also suggests that users are not only interested to see computationally generated poetry as a side effect of their experience, but are willing to have full experiences purely for the computationally generated outcome. Most users mentioned in their interviews that they would use the app again with the sole intention of seeing new bespoke poetry.

Though most participants enjoyed the poetry they received, some did receive poetry without any relatable meaning or coherent structure. This suggests that the prompts used, while appropriate for shorter sets of user answers, were not appropriate when it came to inserting all answers from a user's walk. Future attempts at similar studies should therefore refine the methods of prompting language generation models, perhaps splitting the poetic output into sections, each prompted with a subset of the users' answers.

Chapter 8

Conclusion

As mentioned in 4.2 *Chosen* concept, this project aimed to answer the question: *can computationally generated text improve users' experience when undertaking a Dérive-style walk?* The analysis of user interviews in *Chapter 6 Evaluation* found that users enjoyed the app mostly because of the computationally generated poetry they received after their walk, suggesting the answer to the research question is *yes, considerably so*.

The prototype mobile app developed in this project worked well and all of its intended features were present and able to be used by the participants. The app worked on every participant's phone, and no bugs or issues hindered the user study process.

This project successfully achieved its aims, but the field as a whole has much more that research can look into. It is hoped that this project has contributed to building the foundations of future research into potential use cases for computationally generated poetry and computational creativity in general.

Chapter 9

Reflective analysis

I am very happy with the results of this project and I am proud to have completed a project so much larger than anything I have done before. Though there was certainly room for improvement, many of the challenges faced would not have occurred had my final year not taken place during a pandemic.

Overall, the project was successful and I am both pleased with and proud of the outcome. The exploration gave a fantastic insight into potential use cases and the literature search proved extremely useful in regard to the technical aspects of development and also in helping further my knowledge of the theory which inspired and aided this project. The design stage produced good speculative concepts, which helped me solidify the idea to take forward, and the design of the chosen concept turned out just as I had hoped. Participants were enthusiastic and understood the project well, leading to some brilliant user interviews, in turn leading to clear results I am very pleased with.

Because this project took place during the COVID-19 pandemic, government-issued restrictions hindered the user study. If this project were to be done again, without the current restrictions, a more detailed user study could take place. In this case, I would have recruited more participants (though the 10 recruited in this project enabled a successful result, more participants would have given a broader range of experiences to be discussed) and gone further with the user study. Some users (not all, as observations may have intruded upon and thus altered their experience) would have been observed using the prototype app to gain an insight into their actions, including those they may not have remembered when conducting their interviews. User interviews could have been entirely face-to-face, further encouraging follow up questions and answers in a less formal or structured environment, rather than exclusively using phone calls or written communication.

Had there been no time constraint, I would have liked to dedicate more time to the exploration and development of speculative concepts. While I am happy with the designs presented in *4.1 Exploration*, it could have been helpful to have more detailed descriptions and mock-ups, potentially even working prototypes for each concept. That way, a preliminary study could take place, inviting potential users to criticise and discuss the range of ideas presented, with the best-received concept being taken further into complete development. Similarly, the prototype I developed was designed as a prototype and nothing more. Given more time, I would have improved the app's user interface and developed it to the standard of currently available published apps. This would make the experience more realistic, though no participants commented on either the user interface or user experience of the app.

The project management methodology was followed well, with the majority of my work occurring in the time periods initially planned. Some stage finished a few days early, so the next stage was started before it was planned to. Similarly, some stages finished late, so the next stage begun later than planned. Overall, these inconsistencies between the planned times and the actual times tended to balance out. Due to circumstances not explicitly related to this project, I sought a deadline extension of one week and in just a few days, the work that had been pushed slightly back by the unpredictable timings was completed.

The methodology of the project's prototype development and research methods was followed well and I often found myself referring back to my methodology section and comparing my work to my plan. The detailed methodology initially set out helped my progress and I am glad to have spent the time researching potential methods.

I believe I now have the knowledge to improve upon this project if I were to undertake it again, but I am immensely proud of what I have achieved in this project and what I have learned along the way.

References

- Amazon (n.d.) *What are Alexa voice profiles?* Seattle, WA, USA: Amazon. Available from <https://www.amazon.co.uk/gp/help/customer/display.html?nodeId=GYCXY2AB2QWZT2X> [accessed on 8 April 2021].
- Apple (2021) *Set up multiple users on HomePod*. Cupertino, CA, USA: Apple. Available from <https://support.apple.com/en-gb/HT204753> [accessed on 8 April 2021].
- Auger, J. (2013) Speculative design: crafting the speculation. *Digital Creativity*, 24(1), 11-35.
- Bobadilla, J., Ortega, F., Hernando, A., and Guitérrez, A. (2013) Recommender systems survey. *Knowledge-Based Systems*, 46, 109-132.
- Brooke, J. (1996) SUS: a ‘quick and dirty’ usability scale. *Usability Evaluation in the Industry*, 189.
- Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D. (2020) Language models are few-shot learners [pre-print]. Available from <http://arxiv.org/abs/2005.14165> [accessed 15 March 2021].
- Cabinet Office (2020) *Review of two metre social distancing guidance*. London: Cabinet Office. Available from <https://www.gov.uk/government/publications/review-of-two-metre-social-distancing-guidance/review-of-two-metre-social-distancing-guidance> [accessed 17 April 2021].

- Canary, A. (2019) *How to analyze interview transcripts in qualitative research*. San Francisco, CA, USA: Rev. Available from <https://www.rev.com/blog/analyze-interview-transcripts-in-qualitative-research> [accessed 5 May 2021].
- Caulfield, J. (2019) *How to do thematic analysis*. Amsterdam, Netherlands: Scribbr. Available from <https://www.scribbr.com/methodology/thematic-analysis/> [accessed 10 May 2021].
- Colton, S., Goodwin, J., Veale, T. (2012) Full-FACE poetry generation. In: *Third International Conference on Computational Creativity*, Dublin, Ireland, 30 May – 1 June. Association for Computational Creativity, 95-102. Available from <http://computationalcreativity.net/iccc2012/wp-content/uploads/2012/05/095-Colton.pdf> [accessed 31 March 2021].
- Debord, G. (1955) Introduction to a critique of urban geography. *Les Lèvres Neus*, 6.
- Duffield, E. (2021) *Bespoke for billions: human experiences at scale*. New York City, NY, USA: The Wall Street Journal. Available from <https://deloitte.wsj.com/cmo/2021/03/01/bespoke-for-billions-human-experiences-at-scale/> [accessed 8 April 2021].
- Dugan, L., Ippolito, D., Kirubarajan, A., and Callison-Burch, C. (2020) RoFT: a tool for evaluating human detection of machine-generated text [pre-print]. Available from <https://arxiv.org/pdf/2010.03070.pdf> [accessed on 15 March 2021].
- Encyclopaedia Britannica (2017) *Ramon Llull*. Chicago, IL, USA: Encyclopaedia Britannica, Inc.
- Fowler, M. and Highsmith, J. (2001) The agile manifesto. *Software Development*, 9(8), 28-35.

- Gervás, P. (2000) WASP: evaluation of different strategies for the automatic generation of Spanish verse. In: *Proceedings of the AISD-00 symposium on creative and cultural aspects of AI*, Birmingham, 17-20 April, Birmingham: University of Birmingham, 93-100. Available from https://www.researchgate.net/profile/Pablo-Gervas/publication/228609235_Wasp_Evaluation_of_different_strategies_for_the_automatic_generation_of_spanish_verse/links/00b4952aada6407047000000/Wasp-Evaluation-of-different-strategies-for-the-automatic-generation-of-spanish-verse.pdf [accessed 15 March 2021].
- Harrell, M. C. and Bradley, M. A. (2009) *Data collection methods. Semi-structured interviews and focus groups*. Rand National Defense Research Institution Santa Monica.
- Huang, C. and Kusiak, A. (1996) Overview of Kanban systems. *International Journal of Computer Integrated Manufacturing*, 9(3), 169-189.
- Internationale Situationniste (1958) Definitions. *Internationale situationniste*, 1. Translated by Knabb, K. Available from <https://www.cddc.vt.edu/sionline/si/definitions.html>.
- Izraylevych, I. (2018) 10 strong reasons to choose react native for mobile app. San Francisco, CA, USA: Medium. Available from <https://igorizraylevych.medium.com/10-strong-reasons-to-choose-react-native-for-mobile-app-d7f0ce1a6f4c> [accessed 16 April 2021].
- Jurison, J. (1999) Software project management: the manager's view. *Communications of the association for information systems*, 2(1), 17.
- Kirubarajan, A. and Dugan, L. (2020) Learning to trick humans: evaluation criteria for human-written and machine-generated text.

- Köbis, N. and Mossink, L. (2021) Artificial intelligence versus Maya Angelou: experimental evidence that people cannot differentiate AI-generated from human-written poetry. *Computers in Human Behaviour*, 114.
- Lamb, C., Brown, D. G., Clarke, C. L. A. (2017) A taxonomy of generative poetry techniques. *Journal of Mathematics and the Arts*, 11(3), 159-179.
- Lu, J., Wu, D., Mao, M., Wang, W., Zhang, G. (2015) Recommender system application developments: a survey. *Decision Support Systems*, 74, 12-32.
- Lyons, K. (2020) *Now all Google Assistant devices will be able to tell who's talking to them*. New York City, NY, USA: The Verge. Available from <https://www.theverge.com/2020/6/11/21288359/google-voice-match-assistant-devices> [accessed on 8 April 2021].
- Mann, W. (1982) Text generation. *American Journal of Computational Linguistics*, 8(2), 62-69.
- Manurung, R., Ritchie, G., Thompson, H. (2011) Using genetic algorithms to create meaningful poetic text, *Journal of Experimental and Theoretical Artificial Intelligence*, 1, 43-64.
- McKeown, K. (1992) *Text generation*. Cambridge: Cambridge University Press.
- Misztal, J. and Indurkha, B. (2014) Poetry generation system with an emotional personality. In: *International Conference on Computational Creativity*, Ljubljana, Slovenia, 9-13 June. Association for Computational Creativity. Available from https://www.researchgate.net/publication/274249704_Poetry_Generation_System_With_an_Emotional_Personality [accessed 15 March 2021].
- Moult, M. (2011) Cued recall. *Encyclopedia of Clinical Neuropsychology*. New York City, NY, USA: Springer.

- Nemoto, T. and Beglar, D. (2014) Developing Likert-scale questionnaires. In: *JALT2013 Conference Proceedings*, Tsukuba, Japan, 20-23 November. Tokyo, Japan: JALT. Available from https://jalt-publications.org/sites/default/files/pdf-article/jalt2013_001.pdf [accessed 11 May 2021].
- OpenAI (n.d.) *Playground – OpenAI API*. San Francisco, CA, USA: OpenAI. Available from <https://beta.openai.com/playground/> [accessed 24 April 2021].
- Piper, K. (2020) *GPT-3, explained: this new language AI is uncanny, funny – and a big deal*. Washington, DC, USA: Vox Media. Available from <https://www.vox.com/future-perfect/21355768/gpt-3-ai-openai-turing-test-language> [accessed 14 May 2021].
- Rajcic, N. and McCormack, J. (2020) Mirror ritual: an affective interface for emotional self-reflection. In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, Honolulu, HI, USA, 25-30 April. New York City, NY, USA: Association for Computing Machinery. Available from <https://dl.acm.org/doi/10.1145/3313831.3376625> [accessed 15 March 2021].
- Savage, M. (2021) *Spotify wants to suggest songs based on your emotions*. London: BBC. Available from <https://www.bbc.co.uk/news/entertainment-arts-55839655> [accessed on 15 March 2021].
- Schäfer, J. (2006) *Literary machines made in Germany. German cybertexts from the Baroque era to the present*. Jyväskylä, Finland: University of Jyväskylä Research Centre for Contemporary Culture.
- Schwaber, K. (1997) SCRUM development process. *Business Object Design and Implementation*, 117-134.

- Scott, K. (2020) *Microsoft teams up with OpenAI to exclusively license GPT-3 language model*. Redmond, WA, USA: Microsoft. Available from <https://blogs.microsoft.com/blog/2020/09/22/microsoft-teams-up-with-openai-to-exclusively-license-gpt-3-language-model/> [accessed on 15 March 2021].
- Shi, Y. (2019) *Exploring use cases for an artificial intelligence poet*. MSc. KTH Royal Institute of Technology. Available from <https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1415979&dswid=9771> [accessed 15 March 2021].
- Spotify (n.d.) *Company Info*. Stockholm, Sweden: Spotify. Available from <https://newsroom.spotify.com/company-info/> [accessed 15 March 2021].
- Statista (2021a) *Number of Netflix paid subscribers worldwide from 1st quarter 2013 to 1st quarter 2021*. New York City, NY, USA: Statista. Available from <https://www.statista.com/statistics/250934/quarterly-number-of-netflix-streaming-subscribers-worldwide/> [accessed 15 March 2021].
- Statista (2021b) *Number of monthly active Facebook users worldwide as of 4th quarter 2020*. New York City, NY, USA: Statista. Available from <https://www.statista.com/statistics/264810/number-of-monthly-active-facebook-users-worldwide/> [accessed 15 March 2021].
- Streefkerk, R. (2019) *Qualitative vs. quantitative research*. Amsterdam, Netherlands: Scribbr. Available from <https://www.scribbr.com/methodology/qualitative-quantitative-research/> [accessed 17 April 2021].
- The University of Newcastle (2020) *Research methods: what are research methods?* Newcastle, Australia: The University of Newcastle. Available from <https://libguides.newcastle.edu.au/researchmethods/home> [accessed 17 April 2021].

Trettien, W. (2010) *Fünffacher Denckring der Teutschen Sprache*. Chicago, IL, USA: Electronic Literature Organization. Available from <https://directory.eliterature.org/individual-work/574> [accessed on 13 May 2021].

Woolley, G. H. (1969) Automatic text generation. In: *International Conference on Computational Linguistics*, Svartsjö, Sweden, 1-4 September. Stroudsburg, PA, USA: Association for Computational Linguistics. Available from <https://www.aclweb.org/anthology/C69-3701> [accessed on 15 March 2021].

YouTube (n.d.) *Press – YouTube*. San Bruno, CA, USA: YouTube. Available from <https://www.youtube.com/intl/en-GB/about/press/> [accessed 15 March 2021].

Appendix

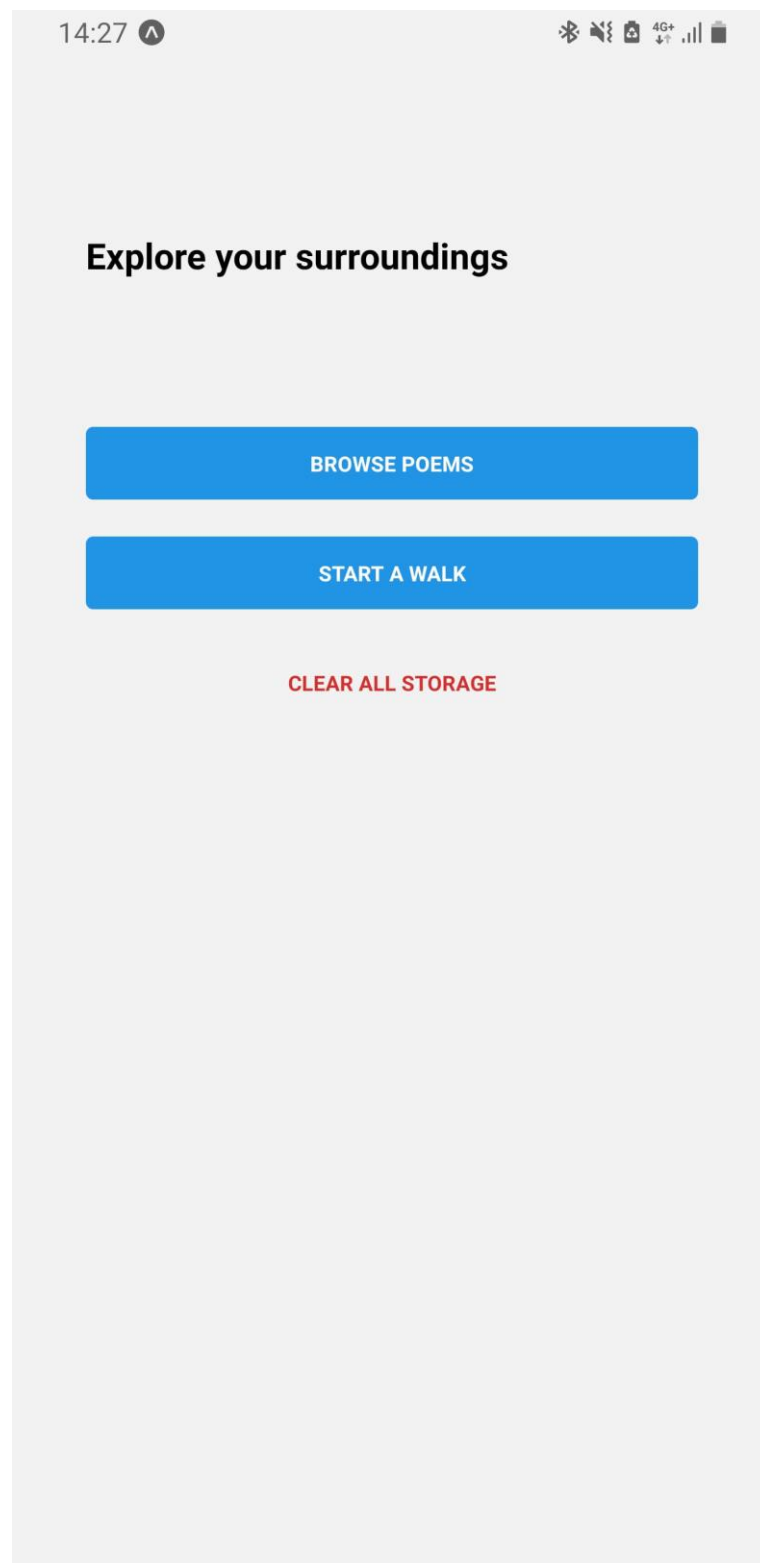


Figure 17: the prototype app's home screen

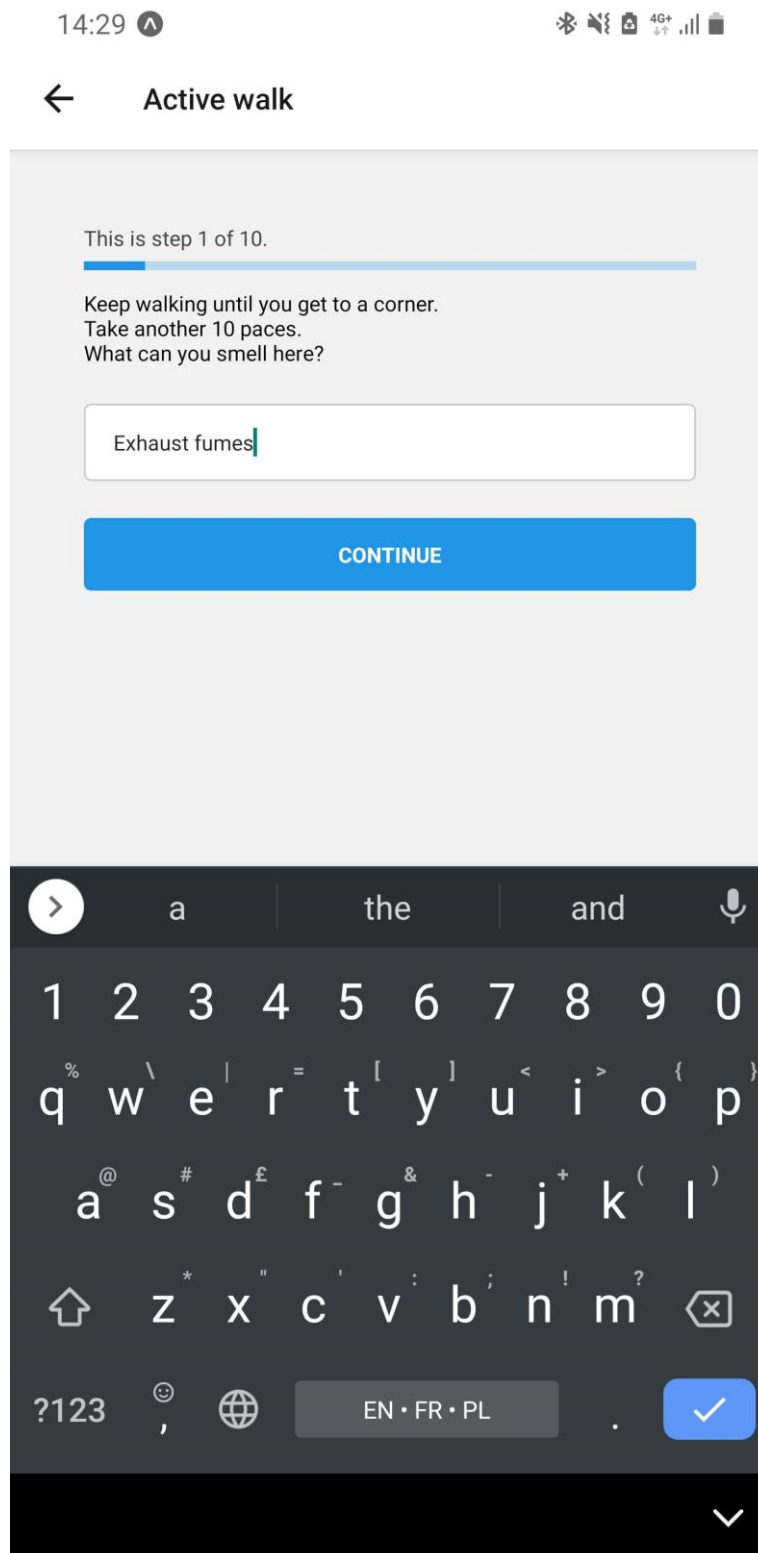


Figure 18: the prototype app's active walk screen

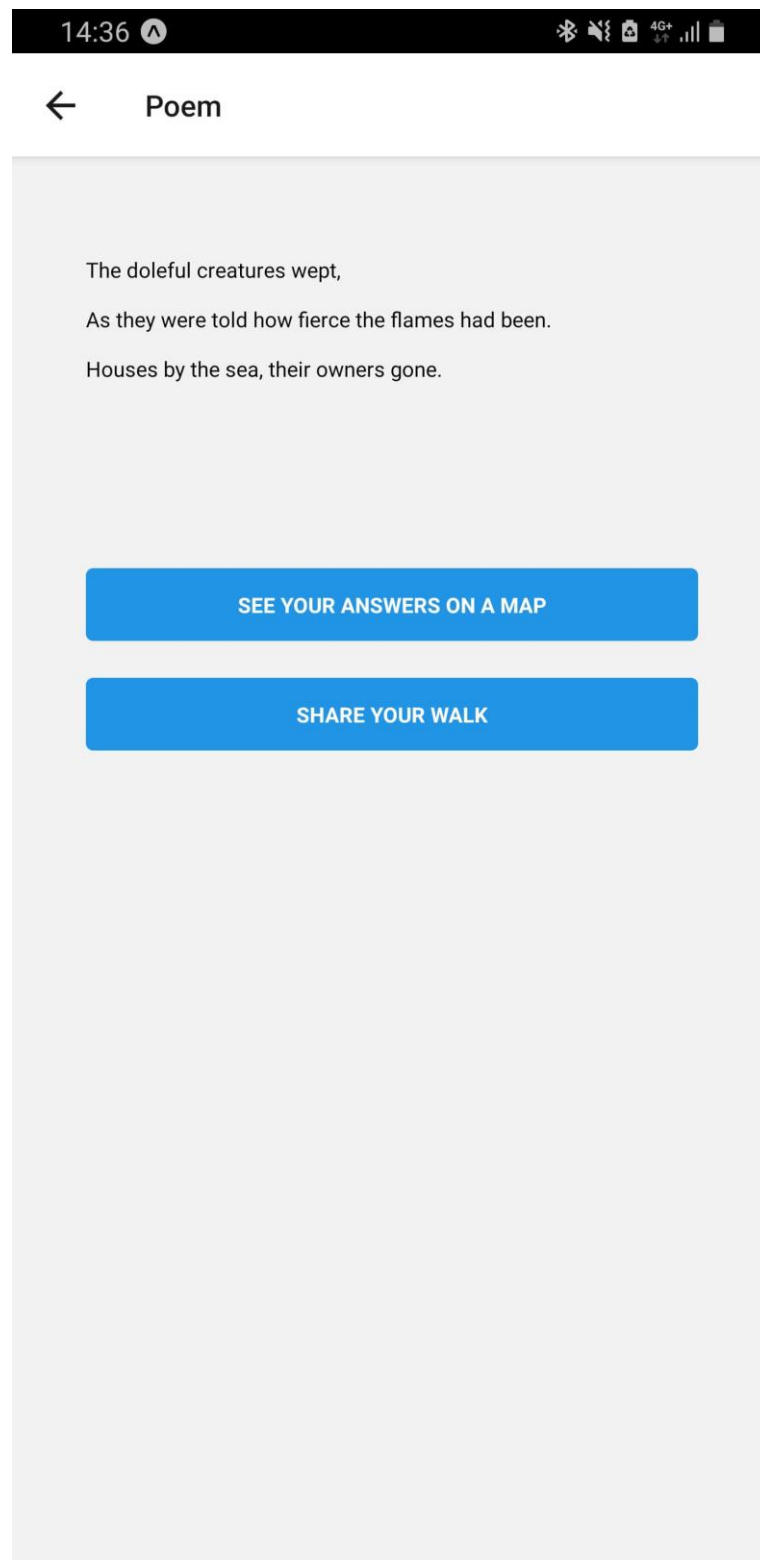


Figure 19: the prototype app's poetry screen

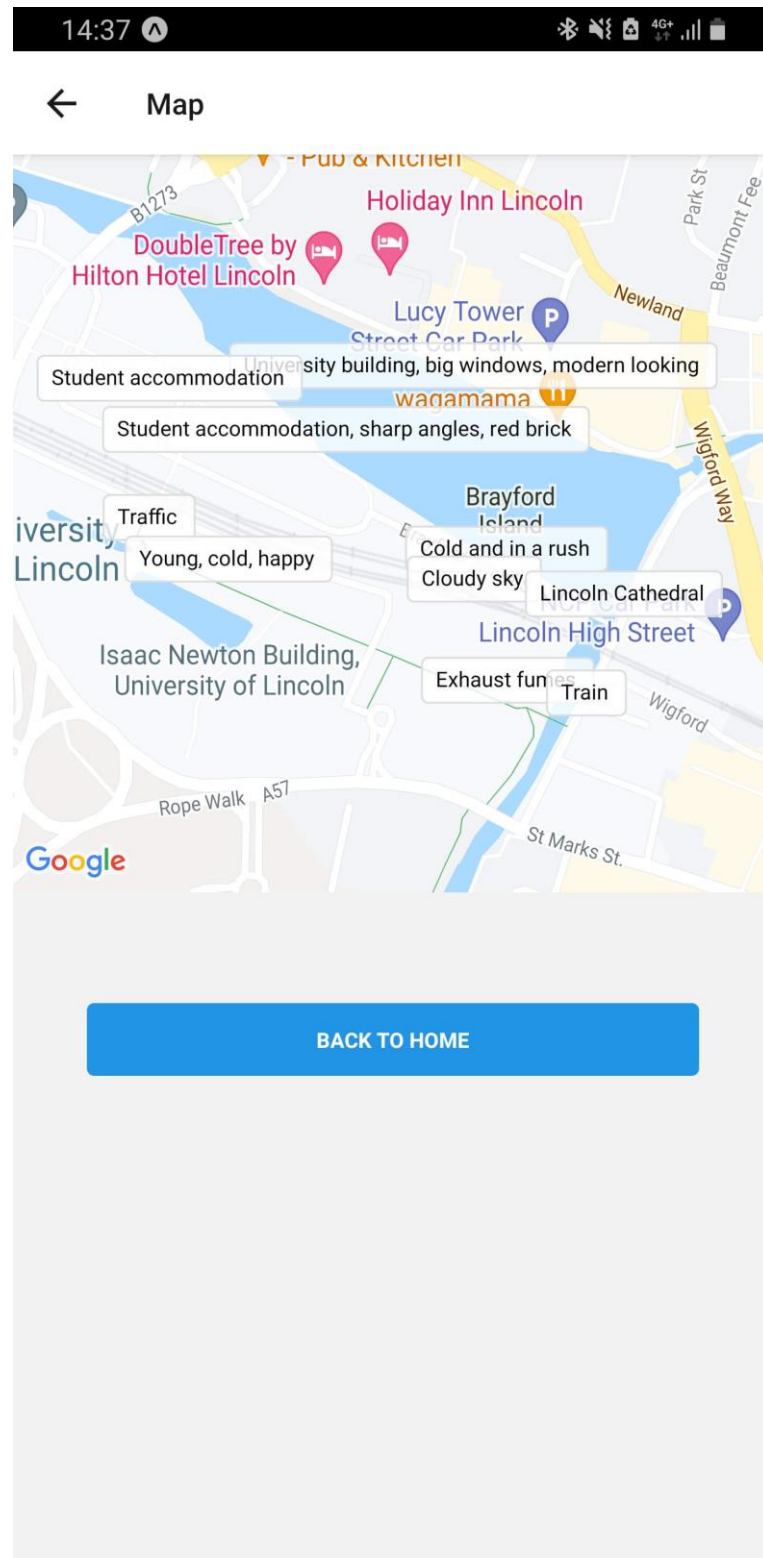


Figure 20: the prototype app's current walk map screen

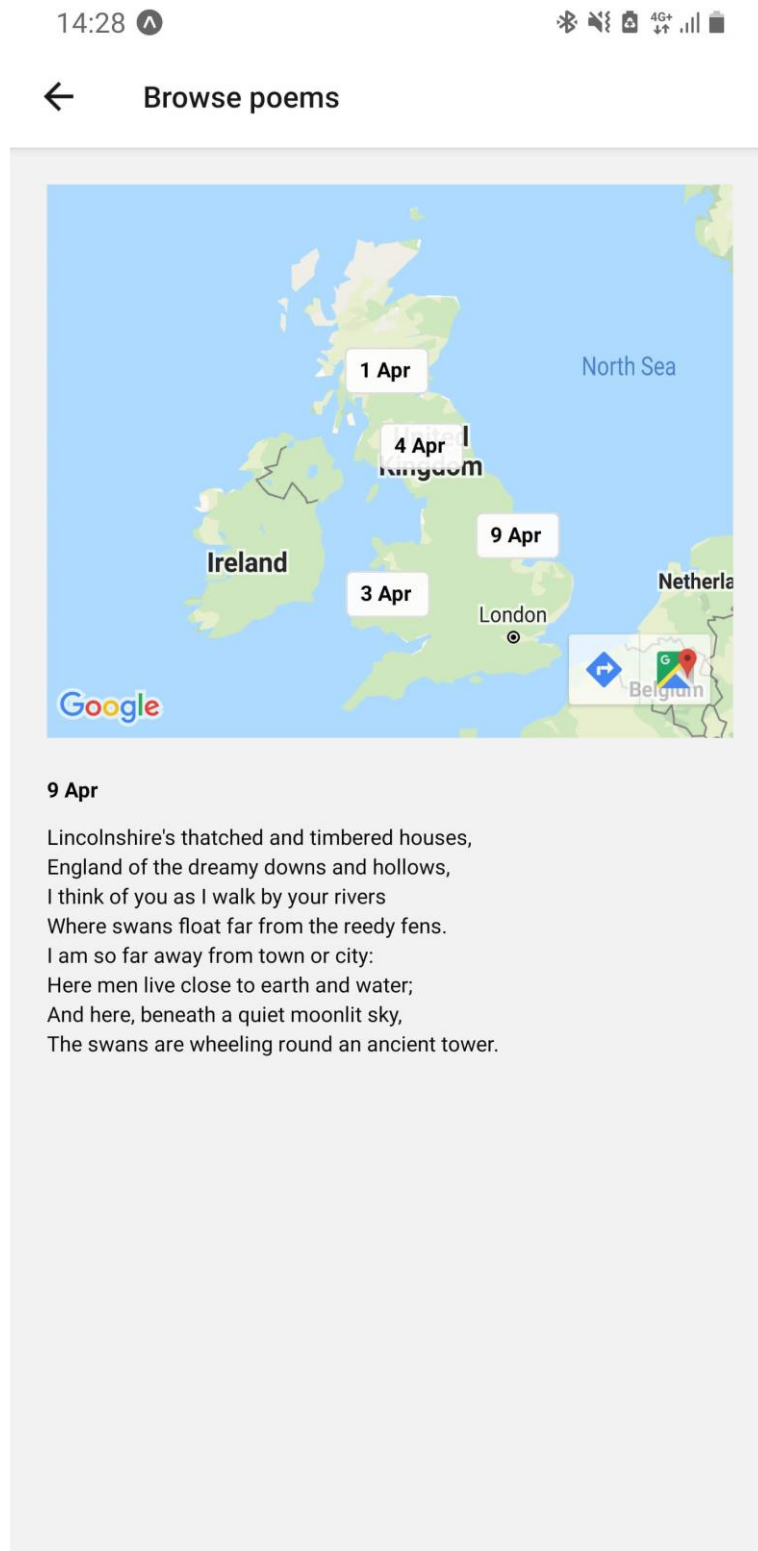


Figure 21: the prototype app's previous walk map screen