

# **Generating Natural Language Descriptions of Rainfall from Radar Data**

**Author :** Daniel Weir C1769632

**Degree :** MSc Computing

**Supervisor Name :** Dr Padraig Corcoran

**Institution :** School of Computer Science and Informatics  
Cardiff University

**Date :** 19<sup>th</sup> September 2022

## **Table of Contents**

1. Introduction
2. Literature Review
3. Methodology
  - a. Radar Data
  - b. Requirements Specification and System Analysis
  - c. Natural Language Generation Tasks
    - i. Content Determination
    - ii. Discourse Planning
    - iii. Sentence Aggregation
    - iv. Lexicalisation
    - v. Referring Expression Generation
    - vi. Linguistic Realisation
  - d. Template
4. Implementation
  - a. Geo-Coding
  - b. Chrono-Requesting
  - c. Precipitation Characterisation
5. Evaluation and Analysis
6. Discussion
7. Conclusion
8. References

## **Abstract**

During this project, I have explored the computer science sub-field of Natural Language Generation (NLG) by creating a system that takes as input rainfall radar data and outputs a generated natural language description of such data. In the course of this dissertation I will review key literature regarding the broad field of NLG and its applications, as well as looking in more detail at weather event specific natural language processing and generation systems. After which, I will detail the methodology of my project and detail how the system works, and evaluating its success. This paper will then conclude with a discussion on the systems implementations and wider utilisations outside its initial scope.

## **Introduction**

The stereotype that people in Britain love to discuss the weather seems firmly cemented in the minds of those abroad and at home. With major domestic news outlets publishing articles, seemingly annually, on alleged studies with claims that we spend “six months of our lives discussing the rain” and “94% of us have talked about the weather within the last six hours”, it is about time that these discussion were automated.

Natural Language Generation (NLG) is the sub-field of Natural Language Processing (NLP) that seeks to transform some set or series of data into a comprehensible and comprehensive “Human Language”. In short, translating the zeros and ones a computer hears into English or Welsh or any other language for people to understand. The aim of this project was to develop a NLG-system that would receive radar data, depicting rainfall over the UK, and would return a Natural Language Description (NLD) that describes where, when, and how much rain had fallen in plain English. Here the scope of this project begins to define what shall and shall not be included. The explicit interest of the project is in the rainfall itself, and no matter the pertinence of the cause, the meteorological explanations for how and why rainfall occurs as it does is beyond the scrutiny of this investigation. Also excluded from the scope is rainfall occurring outside of the UK, which for any Americans among the readership, is the sovereign state consisting of the constituent countries of Wales, Scotland, Northern Ireland, and England. Pedanticism here is necessary to ensure a precise understanding of the margin of this program of study. While NLG can be conducted in a variety of languages, and indeed more research ought to be conducted into the development of systems in minority languages, the linguistic capabilities of the researcher limit the project to only conduct itself in English. It is left to the Discussion portion of this paper to probe the prospects of adapting such a system to Cymraeg or Ghàidhlig or another of the rich and diverse languages on offer

within the UK. It should lastly be trivial, but let's be unequivocal, to state that the scope only surrounds rain events, and other weather phenomena are excluded from this project, though future adaptation to cover such areas of climate interest such as heatwaves or wind speeds will not be excluded from review. This exceptionally narrow scope does create limitations for the outcomes and applications of the project. Restrictions to language mean that understanding of NLG and the wider NLP field is limited to the specific linguistic and grammatical features of English and further understanding of the nuances of text generation in linguistically polarising languages cannot be developed. The narrowness of the accepted input data also means that cross referencing the accuracy of the data is impossible and therefore the credibility of the output data immediately rests on the integrity of the MET Offices data. The data type also limits the direct comparability of much pre-existing literature that doesn't make use of the same or similar data sources.

Justification and indicating the relevance any research is of course a necessity, but Reiter and Dale (1997) point to the specific requirement of justifying an NLG system in particular. Indeed Dale (2020) discusses the view of NLG being the "poor relation" in the field of NLP as the need to represent data in linguistic format may not be evident to all, particularly those who would argue that visually representing data is more efficient and effective. To spend days producing a system that might lie dormant on a shelf never to be enlisted would be a great waste of time, so why should this system be created?

Firstly, accessibility. While visualisations of rainfall data can be efficient in disseminating data to a wider audience, that efficiency is predicated on a variety of assumptions. The most blatant assumption is that the audience can see the data. Visual impairment of a range presentations can exclude the data from being understood by the audience. Total or near blindness can exclude an individual from being able to interpret the data entirely, and while a written description of the radar data cannot immediately solve this issue, it is the first step in producing auditory or braille broadcasting of the information. The radar images employment of colour scaling might also exclude those with colour blindness, so producing a linguistic description of the data makes it more available. Textual data is also easier to disseminate to an array of sources individual might use to access data. Poor internet connectivity or lack of access to adequate technological equipment limit the ability of those audiences affected by such restraints to acquire the data in the visual format. Reducing the bandwidth of the data and the ability to print it requiring less ink are benefits of data-to-text systems. The current format of the data also requires users to be aware of specific details

pertaining to how the images are interpreted. This will be discussed in more detail in the Methodology section of this paper but for now it suffices to say that readers who lack specialised familiarity with the data would be unable to yield legitimate conclusions from it, and therefore an aim of this project is to remove the necessity of users needing a key to understand the data.

Secondly, productivity. An NLG system in this case will be able to much more quickly evaluate, and compile an accurate synopsis of the data than a human performing the same task. Here I must review the ethics of a system that would render an individual obsolete, however I conclude that this is a task that would not be categorised as the main focus of any individuals job role and thus automating it proffers assistance to those who might have need of a system rather than threatens their economic security and employment tenure. Reiter et al (1995) also note that such systems ensure conformity of structure and content in the summary of information which is a benefit when continuity is a necessity of the user. But why should this specific system of describing past rainfall from radar data exist? In a wider context, a system that can interpret an array of radar images and data not exclusive to rainfall can be of benefit to many professions whose analysis of the weather is a secondary or tertiary task that may distract them from their main position. Here, think of police investigators who may need to review the rain conditions around the time of a road traffic collision as per investigation guidelines. (It must be noted that the use of a system in such a way would throw up legal implications and responsibilities if the description were used as evidence in court). In relation to more contemporary events, picture that a system such as this could be used to describe differences in precipitation rates between the recent hot and drought ridden summer and summers previously where the uniformness of the descriptions will contrast the lack of consistency in the rainfall volumes and convince certain spheres of twitter and the general public that the weather events and lack of rain being experienced more frequently is very much abnormal and indeed, terribly concerning.

This paper begins with review of existing literature regarding the field of NLG. The review will look at wider research regarding NLG as a whole and how NLG systems may be implemented and created, before looking more in depth at systems and research that are specific to creating NLDs of weather events. The review will also look at some non-NLP specific programming challenges and techniques that are necessary to complete the system. Following on from the review segment will be a chapter on methodology which will start by offering information in regards to the data sources used to implement the software before

detailing how the system was assembled, with a run through of how it work. This chapter will conclude with an analysis of how the system performs. The Discussion section will look at the implementation of the system and how it can be adapted in the future to serve a wider purpose whilst also looking at improvements that could be made.

## Literature Review

This review will look at literature that covers 3 distinct areas of importance for the undertaking of this project. Firstly, by looking at the wider field of NLG and NLP I will develop and convey an understanding of the key tasks that will be necessary to comprehend and perform in order to develop my own working NLG system. Secondly, papers that are specific to the creation of NLDs of weather events will be explored to gain an understanding of the subject specific struggles and constraints in developing a computer system that can accurately convey weather details to people in human languages. These 2 topics shall make up the bulk of this review segment. The third area of review will be brief, but will look at areas of interest such as software development processes, that will be important to the wider project.

The first piece of literature that must be reviewed when conducting any form of work with regards to NLG is Reiter and Dale's 1997 paper, "*Building applied natural language generation systems*". Its importance and relevance to the field is evidenced by its 2837 citations listed on Google Scholar and all literature reviewed here, post its publication date, cites their work. At 25 years old, the concern that it is outdated is valid (with the reference to the antiquated "teletext" system in section 2.2.1 testimony to this), however I would argue that despite the advancement of technology since, the key tasks outlined within it remain crucial to the development of NLG systems today, and citations from as recently as June 2022 would seem to concur with this verdict.

The paper introduces two categories of chores that an NLG system can be used to automate. The first, using an NLG as an authoring aid to automate the writing of a routine or simple document, such as writing letters to customers (Coch 1996). The second, converting some technical representation of data into a human language representation, such as describing weather forecasts (Goldberg et al 1994). The aim of this project aligns with the second category, where my aim is to produce a system that converts a radar image that requires technical knowledge to interpret. (An overview of the technical knowledge I am aiming to circumvent is provided in the Methodology section) .

They go on to argue that the task of creating an NLG system can be broken down into 6 basic activities. These are surmised thusly;

- Content Determination:

The task of determining what information is included in the output text.

- Discourse Planning:  
The task of determining the order of the information and the structure of the output text.
- Sentence Aggregation:  
The task of combining sentences to assist in the fluency of an output text.
- Lexicalisation:  
The task of choosing which words to use to express specific concepts and convey ideas.
- Referring Expression Generation:  
Similar to lexicalisation, this task determines how the system will refer to entities included within the text, typically either by name and proper noun, or using personal pronouns like “he”, “she”, “they”, or “it” for inanimate objects.
- Linguistic Realisation:  
The task of ensuring the system outputs is grammatically correct. For example using the correct tense of verbs for past or present events, and the capitalisation of proper nouns.

It may appear I have dedicated an excessive portion of this literature review to describing the contents of this paper. However I believe it important to offer a sufficient overview of these tasks as they have informed a large part of my Methodology and therefore it is essential to provide them as reference here to both critically analyse that which has been an underlying blue print of my project, and provide Reiter and Dale the just recognition they deserve for their contribution to the field.

While the aim of the project does not fully align with the aim of the system investigated by Coch (1996), the techniques utilised for producing NLG text, and the processes used in assessing the outputted text, are both of keen interest to this project. The Automatic Hybrid Generation method for creating text details a system that utilises a template approach that returns a filled out customer letter based on specific input details. This system does require human validation of the auto-filled gaps which is not utilised in this project, however the human input of some details such as addresses is adaptable to the goals of this dissertation. More importantly however, the paper outlines several key areas that it investigates to assess the success of the system. These are as below;

- Correct spelling



- Good grammar
- Comprehensiveness
- Rhythm and Flow
- Appropriateness of the tone
- Absence of repetition
- Correct choice and precision of the terminology used.

These will be used as a reference during the Methodology section as a guide to good practice in making certain decisions regarding the template formulation for the NLG system, and then in the Evaluation section to assess the programmes ability to produce “good” text outputs.

Goldberg et al (1994) highlight the ever-growing amount of data that is accessible when it comes to weather forecasts. They discuss the benefits of an NLG system that is able to more quickly evaluate and extract information from a large collection of data and create an accurate weather forecast quickly and efficiently. However they also raise concerns around accountability of these systems in cases of hazardous weather. The legal implications of using automated systems has already been alluded to in the introduction of this paper. Take for example, a traffic collision investigator who was to submit as evidence output produced by the system created in the course of the project as to the rain conditions or lack thereof that contributed to a crash. Where for does the accountability for the accuracy of this data lie? In the judgement of the investigator, or in the system itself? If the data is inaccurate, who lies at fault? In Goldberg et al’s paper they highlight the specific desire for such systems to be used to automate routine roles of forecasting “so [forecasters] can concentrate on the scientific questions”. Here resolves the ethical issue. Any such system should be used to provide an initial indication to those using it as to which direction they ought to be directing their attention their own further investigations on which civilians rely on the accuracy and accountability of their findings.

They go on to detail the 3 stages they identify the FOG system cycles through to produce its data-to-text output. These being;

1. Data Extraction
2. Conceptual Processing
3. Linguistic Processing

In relation to this project, data extraction pertains to the analysis of the radar data and extracting the pertinent information using image processing techniques. Using the time series

radar maps depicting rainfall data and the processing programme, data extraction will withdraw from the input the key data points outlined in the methodology section.

Conceptual Processing as it pertains to this project involves reducing the amount of data to specific key points and applying to these specific conditions in how they ought to be presented. In the case of the radar system, this involves reducing the amount of distinct rainfall amounts to the most frequently occurring precipitation rate, and mapping the location of the rainfall to a certain parameter area, e.g. town/city level, or national.

Linguistic Processing in their paper consists of text determination and text realisation. These are a simplification of the 6 NLG tasks Reiter and Dale listed that will be utilised instead in the course of this project.

They go on to detail the Sublanguage analysis conducted in developing a corpus to draw from when determining linguistic word choices for describing weather events. They note that weather forecasts use a particular sublanguage in both English and French that makes use of differing tensical and syntactic word choices to present data. Indeed these linguistic choices and methods for approaching how to make such choices is covered extensively by Reiter et al (2005). For the purposes of this project, sublanguages considerations are made under the NLG tasks of lexicalisation and linguistic realisation subsections in the Methodology chapter and utilise the “National Meteorological Library and Archives-Fact sheet No. 3-Water in the atmosphere” as the basis for those linguistic word choices aforementioned.

In their paper “Generating Spatio-Temporal Descriptions in Pollen Forecasts” (2006), Turner et al detail the steps they take in developing an NLG programme that describes pollen forecasts for Scotland, building on the frameworks laid out by Reiter and Dale (2000) and Sripada et al (2003) to produce a prototype. They succinctly surmise the two main tasks that must be completed when “automatically generating spatio-temporal descriptions”. They state, in more convoluted terms, the first task is identifying and extracting the data necessary to satisfy the requirements of the description. In their case, that would be the levels of pollen, where those levels are, and when. The second task is determining the most accurate way to describe the location. They noted differences in how individuals may categorise the location of different cities and points of interest in Scotland which means using sweeping geographical areas such as “North East” or “South West” could result in issues in comprehension on the human end and result in poorer understanding of the generated

descriptions thus sacrificing the efficacy of the prototype, damaging its potential applications. These are important aspects that must be considered when building an NLG system and will be discussed in the methodology chapter in addition to suggestions to attempt to resolve them.

Davy et al (2008) also discuss the nuance in choosing spatial determiners for NLG systems and note 4 key areas that influence the specific syntactic choices made when specify spatio-descriptors. These being altitude, coastal proximity, population, and direction. These are of specific interest in their research regarding frost and fog forecasting systems however it is worth noting them here as discussion surrounding the population and direction categories will be considered later in choices made to make the system more effective and efficient, and how changes could be made to adapt and improve the system overall. Determining altitude of regions from the radar data that will be utilised by the system is possible but lends itself more into comprehending rain events and why they occur in the way they do which, as already stated, is out of the initial scope of the project. The main point of interest from this paper comes from its categorisation of 4 potential types of weather descriptions. These being;

1. Overview – which describes a general pattern in the weather.
2. Time Series – which describes the state of a specific weather parameter over some time frame.
3. Stationary – which describes weather events at a specific time within the data.
4. Nonstationary – which describes how a localised weather event develops.

These categorisations are utilised in the research to reduce corpus texts to the most significant details to make content determination more efficient. The system being produced within this project will utilise the first 3 categorisations in an adapted format. The fourth categorisation is quickly omitted as the data set utilised in the course of this assignment is historical data and under methodology is reduced to hour long intervals so limits the necessity of investigating nonstationary events. Of the 6 details that the corpus texts in this paper are reduced to with these categories, this investigation requires analysis of 4. The first, Time Period. This shall be the easiest for our system to determine. The second Trend. In the radar maps, this will be the precipitation rate. The third, Area. This will be covered in more detail under lexicalisation when discussing spatial quantifiers. The last, Frame of Reference. Davy et al make reference to the 4 spatio-descriptors aforementioned however in the course of this project this would more accurately be reflected by the location areas being quantified on a city or country level. The remaining 2 are trivial for this project as they concern “parameters”

i.e. weather events, and “main verb” and these are already known to be rain and rained respectively. Thanks to the explicitness of their paper, content determination occurring during methodology was made easier later on in this project.

Ramos-Soto et al (2014) also discuss the automatic generation of textual weather data. In common across all 4 of these papers is the use of weather forecasting data as the input data in producing these data-to-text descriptions of a broad array of weather events. Here the specific objective of the project asserts itself. In the course of accumulating literature, at no point was it evident that a system had been designed that would utilise pre-existing radar data as its input to generate historical descriptions of rain events. Furthermore, all were reliant on pre-determined forecasts or utilising meteo-statistical modelling to ascertain future weather, none required the use of image processing techniques to interpret graphical recorded data. As such their technical solutions to extracting the data from the source are omittable however the ways in which they discuss the mapping of numerical data to a meteorological lexis is of significant interest and convenience as it offers an adequate starting framework for the system to be produced.

Having completed the most significant part of this literature review it is now important to briefly look at a few other pieces of literature that will inform some of the process used in conducting the overall project.

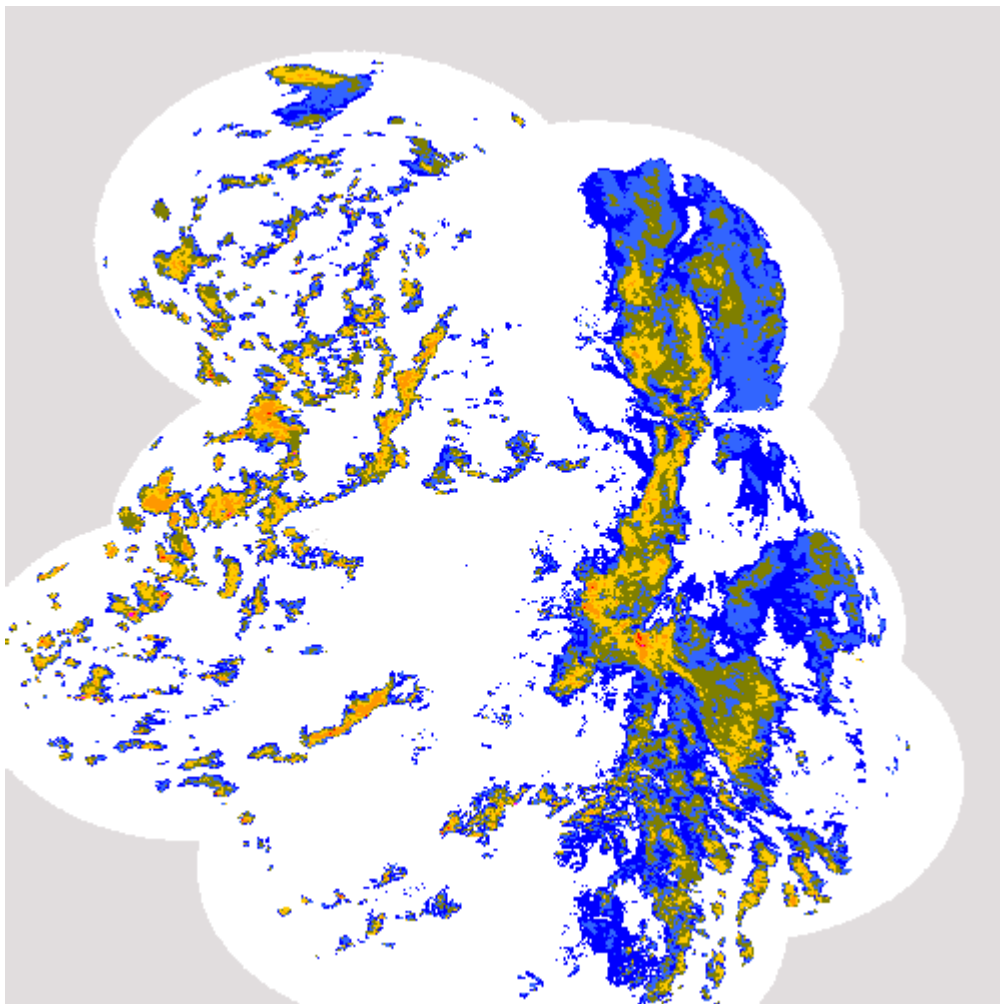
LaPlante (2017) and Sommerville (2005) keenly detail how to adequately determine the system requirements prior to engaging in product development to ensure the key tasks of the project are adequately outlined. Cohn (2004) then proffers great detail on the creation of accurate and efficient user stories that will lend themselves handily in two key aspects of this project. Firstly, in succinctly outlining the tasks the desired system must perform, as outlined by the system requirements adopted as previously informed, to act as a checklist in the development stages to ensure all demands of the project are met. Secondly, in applying these user stories to test case scenarios for the evaluation and analysis of the success of the end product. Here concludes the literature review.

## **Methodology**

In this section I will detail the pre-programming exercises that were conducted to determine the form the system would take on to achieve the goals of the project.

### **Radar Data**

The Met Office DataPoint allows individuals to access and download Met Office Data to be utilised in projects and innovations without restriction. It is from this service that the input data for this system is sourced. An example of the radar maps is displayed below in Figure 1.



*Figure 1 - Radar Map of Rainfall at 11:45, 6/12/2021*

The test data used for this system is a series of Radar images taken between 6/12/2021 and the 17/12/2021 from the MET Office and provided by Dr Padraig Corcoran. During this time MET Éireann had named Storm Barra sweeping across Ireland and the UK. This storm

and the associated rain conditions is well represented within these images and lends itself to the project in offering a rich and rain full data set to test the end system.

As aforementioned in previous sections, the aim of this project is to produce a system that can provide a human language interpretation of the radar map of rainfall to circumvent the technical knowledge needed to understand it so as to make it more accessible to a general audience. The insights required that I aim for the system to perform are;

- Longitudinal and Latitudinal area:

The radar images are bounded between the coordinates  $48^{\circ}$  to  $61^{\circ}$  North and  $12^{\circ}$  West to  $5^{\circ}$  East but this is not immediately evident. They also are not overlaid onto a UK map. This is the first technical information the system will deduce for the user by rendering the missing spatial data into English.

- Precipitation Rate:

The images render the volume of rain as coloured pixels, with each of the 8 potential colours representing a different density of precipitation. (I omit the ninth colour which depicts a lack thereof of rain as we are uninterested for the purposes of this project. A tenth colour depicts the areas within the images in which the precipitation data is out of scope of the radar systems) . Here the aim of the system is to convey these varying precipitation rates in a way that voids the need for an individual to discern the key for each colour.

- Temporal Relativity:

The time the images are taken can be recorded in a variety of ways depending on how the information is accessed, and may not necessarily be recorded at all. A system was developed to standardise how the images were accessed and time stamped and is demoed but this is out of scope of this project. Here the main task of the system is to combine the 4 quarter hourly images and offer a temporal description to the user to prevent potential confusion.

### Requirements Analysis and System Specification

The first task that had to be completed in this project was an analysis of the above aims and goals to lay-out the desired systems requirements and specifications. This is essential as it provides a checklist of tasks the system must be able to perform to satisfy the users' needs and assists in keeping the project on track so unnecessary "add-on" system performances aren't developed before essential aspects.

Using Sommerville (2005) as a frame of reference, requirement specification can be broken down into 3 sub-categories;

- User Requirements:  
Descriptions of the services the system should provide in simple terms. Here I will enlist User Stories to specify these requirements.
- System Requirements:  
Analysis of User requirements informs the System requirements and offer structured and precises descriptions of the systems services and constraints.
- Design Specifications:  
Follows on from system requirements, Design Specifications offer specific descriptions as to the technical details that the developed programs will need to follow and conduct.

Now I list the user stories I detailed, with bullet pointed annotations on the tasks necessary to achieve them.

- 1 As a user, I want to receive information in natural language only so that I can easily understand it.
  - Remove data described in technical terms (as previously described).
  - Describe precipitation rate naturally.
  - Describe spatial data naturally.
  - Describe temporal data naturally.
  - Provide linguistically correct and easily understandable output.
- 2 As a user, I want to receive precipitation rate (PR) data so that I have a sense of how much rain fell
  - Provide NLD of how heavily the rain fell (PR in mm to some NL mapping).
  - Provide NLD of how much area was covered by the rainfall.
- 3 As a user, I want to receive spatial data so that I know where it rained.
  - Provide NLD of where it rained. (Spatial Data relating to towns, cities, countries.)
- 4 As a user, I want to receive temporal data so that I know when it rained.
  - Announce the time that the radar map shows information for.
  - Announce day, month, and year date that the radar map shows information for.
- 5 As a user, I want to choose a date and time so that I can read the rainfall information for the specific time frame I desire.

- Be able to select a day for which to receive rainfall data.
  - Be able to select a time for which to receive rainfall data.
- (These will be dependent on what data is available to be utilised)
- 6 As a user, I want to choose a location so I can read how it rained there.
- Be able to search a location to receive rainfall descriptions on the pre-specified time.
- 7 As a user, I want a summary of the map data so that I can have an overview of the rainfall over the whole UK.
- Provide an NLD of the rainfall from any radar map over the whole area.

These 7 cases provided me the framework to lay-out the key tasks of my project moving forward. Later on we will see how they were used within my chosen NLG framework to inform the content and several other key decisions besides that had to be made.

The bullet point annotations provide more or less the system requirements, and lend themselves as the starting points for the design specifications. From these we will derive an overview of the test cases I will use later to determine the success of the system.

#### Design Specifications:

1. The system will describe precipitation rate in NL as determined by pre-set mapping.
  - 1.1. If multiple precipitation rates are present in a location, the system will statistically analyse and extract the most common precipitation rate.
  - 1.2. Where two precipitation rates are dominant, the higher precipitation rate will be used, as this is more impactful and therefore more important to convey.
  - 1.3. Where precipitation rate is null, the system shall return “It did not rain”.
2. The system will use NL to describe spatial data in pre-determined regions.
  - 2.1. Where rain occurs over large areas, the system will use a pre-set region list to describe the location of the rain.
  - 2.2. Where a user inputs their own location to receive data for, the system will output that location in NL.
  - 2.3. Where a user inputs an unknown location, or a location out of scope, the system will return an error message.
  - 2.4. The system will not take as input, or output, longitudinal or latitudinal coordinates.
  - 2.5. Where multiple location descriptors could be used, the system will output the largest distinct area. I.e. opt for town name over larger county area or smaller ward name.



3. The system will output temporal data in NL.
  - 3.1. The system will combine 4 quarter hourly radar maps starting at n:00 and ending at n:45, and describe this time period in some determined NL format.
4. The system will return in NL an approximation of the rain coverage of an area.
  - 4.1. The system will calculate as a percentage the level of rain coverage experienced over an area.
  - 4.2. Using a pre-set mapping, the system will convert the rain coverage percentage to a NL summary.
5. The system will allow users to select a day, and time, to view data for.
  - 5.1. The system will be pre-loaded with radar maps for a selection of days that the user will be able to choose from.
  - 5.2. The user will be able to select a day of the pre-loaded data to see NLD of.
  - 5.3. The user will be able to select a time from the chosen day to see an NLD of.

Many of the bullet points have been condensed into one over-arching system requirement with the relevant design specifications listed below them. With these in mind, we can continue to the main tasks involved in the creation of an NLG system, ensuring to refer back to these requirements to ensure the goals of the project are achieved. The success of the system in attaining these specification will be measured using test cases that will be seen in detail later on. A brief overview of some of those test cases are as below;

- Test a user can input a city name within the UK and receive an NLG output.
- Test the system can identify what level of rain occurs.
- Test the system can take as input any time within a certain 24 hour period.
- Test the system recognises areas out of scope.

### NLG Tasks

Now we have a thorough overview of the tasks the system must perform, we can begin the task of creating template that will satisfy all specifications relating to generating NLDs of the data to output. The following phases of the methodology concern themselves with the 6 sub-tasks outlined in Reiter and Dale (1997), as previously discussed in the Literature Review Chapter to produce the desired template the system will use to succinctly and accurately execute these aims.

## Content Determination

The first task is determining what information that can be extracted from the radar must be included in order to satisfy the system requirements of this project. I had 3 basic questions that the system had to be able to answer;

1. When did it rain?
2. How heavily did it rain?
3. Where did it rain?

The first question was the simplest. I had to determine exactly what time frames to use to describe when the rainfall occurred. We have as input quarter hourly images of rainfall however I decided a more intuitive time frame would be hourly. As such, I chose that each output would include an hour long time frame, starting from an “o’clock” and ending on the next “o’clock” as this is more straightforward and easier to understand. (For details on the restrictions and complications that arose from this, please see the Results chapter.)

The second question was also easy in terms of deciding the output content. The main decision to take here is whether to include in output texts that it did not rain in certain locations. The purpose of the project, as made clear in the title, is to generate descriptions of rainfall therefore to describe a lack thereof of rainfall is beyond the initial scope. No significant amount of time would therefore be dedicated to describing how it did not rain, except for the system to output “It did not rain” if the input images did not contain evidence of rainfall. Beyond that statement, no content exclaiming aridity or temperature was to be included in the output text.

The final question proposed the most difficulty in determining content. As the images cover a large geographical area, the choice of locations to include in the end text was vast. Here, two main choices were made as to what ought not to be included in the text. Primarily, rainfall occurring over sea would be excluded. This was a practical choice with 2 motivations. Firstly that the purpose and justification of this system as previously detailed sees no need for the system to describe rainfall over the sea as it is of little applicable use beyond maritime industries. Secondly, describing the location of rainfall over sea is limited to using longitude and latitude coordinates which undermines the aim to remove the need for a user to have technical knowledge of these. The secondary choice was to not include a description of rainfall over Ireland. While this system is capable of doing so, and will provide a description of rainfall over specific towns in Ireland when requested by the user however this isn’t

recommended, it will not provide an overall description of the rainfall over the entire country. This is for the reason that the data is taken from the UK Met Office and data from Met Éireann is not used. As an act of caution, I would not describe rainfall over Ireland using UK data as it is collected using weather stations in the UK only and therefore the accuracy of the rainfall data over Ireland cannot be assured. As such, the content determined to be necessary for inclusion in the output text was locations existing on land, either in Wales, Scotland, Northern Ireland, or England. Determining the smallest size of location to include is discussed later.

### Discourse Planning

There were 6 potential orders for which the content data could be structured. Examples of each are shown below;

1. It rained this much. It happened here. The time frame was this.
2. It rained this much. The time frame was this. It happened here..
3. The time frame was this. It rained this much. It happened here.
4. The time frame was this. It happened here. It rained this much.
5. It happened here. The time frame was this. It rained this much.
6. It happened here. It rained this much. The time frame was this.

While all six of these structures convey the same key information, the order of that information infers different prioritisation of each data point. The rainfall description is prioritised first in sentences 1 and 2, second in 3 and 6, and last in 4 and 5. Similarly the order of prioritisation of location and time frame can be seen in the remaining sentences. Choosing which structure to adopt is dependent on determining which is most crucial, and here the need to dispose of structures 3 through 6 is self-evident as the priority of this project and the system is to describe rainfall. Between structures 1 and 2 I then chose to adopt structure 1 and prioritise the time frame last. As seen in the system specification the system takes as input a time frame and therefore the user already has the temporal data, and to include it is almost redundant. In the “Referring Expression Generation” section of this Chapter, I will discuss methods considered for resolving this redundancy. The lack of detail in each of these sentences, (this, this much, and here over an actual time, amount, and location) will be expanded upon in the “Lexicalisation” section.

## Sentence Aggregation

The above examples of sentence orders are somewhat unnatural to read with little fluency. While it cannot be contested that the relevant information is presented clearly and concisely, it is jarring and off putting to read. Sentence aggregation overcomes this by combining the sentences using clauses and conjunctions to output more natural text. Whilst linguistic explanations could be employed here to analyse how best to combine these sentences, it is sufficient to rely on intuition. I combined the sentences to become “It rained this much in this location between these times”. The only linguistic analysis necessary to recognise here is that these sentences have been combined using the prepositions “in” and “between”. This will be pertinent to the discussion coming next under “Lexicalisation”.

## Lexicalisation

The task of lexicalisation was split between 2 main duties. The first, determining a lexis for describing the rainfall. The second, determining how to describe the location of the rainfall. Additional smaller tasks included lexicalisation of the rain coverage and finding a suitable way of denoting the time frame in NL from the many different possible means.

The first task was the biggest and most time consuming. It revolved around finding an NL replacement for the demonstrative phrase “this much” seen under “Discourse Planning”. In Figure 1 below we can see the categories for precipitation rate and their corresponding colour designation on the radar maps. The task of lexicalisation here was outlaid in the technical requirements review, and is to convert these precipitation rates into a natural language description of rainfall.

•	0.01 - 0.5	Grade 1
•	0.5 – 1	Grade 2
•	1 – 2	Grade 3
•	2 – 4	Grade 4
•	4 – 8	Grade 5
•	8 – 16	Grade 6
•	16 – 32	Grade 7

Table 1 showing the amount of rainfall in millimetres and the corresponding colour it is depicted by within the radar images.

The most obvious way to represent these categories linguistically rather than pigmentationally is to simply use the numbers given to us. For example, “It rained between 1 and 2 millimetres” or “It rained more than 32 millimetres”. While this would portray the information accurately, it does little to convey to the reader a NL understanding of what that volume of rain means to the general public.

The Met Office offers its own synoptic definitions of rain<sup>2</sup> in regards to precipitation rate per hour. These are as follows:

1. “Slight” rain – { <0.5mm/h }
2. “Moderate” rain – { 0.5mm – 4mm/h }
3. “Heavy” rain – { >4mm/h }

Here we can see that segmentation has occurred and its two sub-tasks, as attributed by Miller and Han (2001), clustering and classification have condensed the 8 categories into 3 smaller categories. While these NL descriptors of rain provide a concise overview of precipitation rate in easily understood terms, they sacrifice more in-depth understandings of the data by compressing 7 available data representations into 2. To counteract this I de-clustered categories 2 & 3, and using as a guide the attributed NL descriptors “moderate” and “heavy”, defined the following categories as so;

1. “Slight” rain – { <0.5mm/h }
2. “Lightly Moderate” rain – { 0.5mm – 1mm/h }
3. “Moderate” rain – { 1mm – 2mm/h }
4. “Firmly Moderate” rain – { 2mm – 4mm/h }
5. “Heavy” rain – { 4mm – 8mm/h }
6. “Very Heavy” rain – { 8mm – 16mm/h }
7. “Incredibly Heavy” rain – { 16mm – 32mm/h }
8. “Extremely Heavy” rain – { >32mm/h }

It is worthy of note that in original drafts rain in excess of 32mm/h was originally classified as “Violent rain”. However after further investigation, it was discovered consensus dictates that violent rain requires precipitation rates in excess of 50mm/h, and a limitation of the data means I am unable to accurately describe weather events as such without additional input.

The addition of adverbial modifiers is simple and offers a wider range of categorisation for rainfall. While it may be difficult to see how they offer much clarification individually, I argue that their worth in disambiguating the original descriptors comes when comparing locations that before might have had the same NLD, i.e. using the original 3 descriptors, “It rained heavy rain in Cardiff and Edinburgh” appears to convey that Cardiff and Edinburgh experienced similar levels of rainfall. In comparison, “It rained extremely heavy rain in Cardiff and It rained heavy rain in Edinburgh.” informs us that the level of rain experienced over Cardiff was double that of Edinburgh.

The next lexicalisation task was determining how to refer to areas experiencing rainfall. For when users do input their own locations to receive more specialist spatial data, all this task requires is to use their input. The main issue arises when trying to determine how to offer an overall description of rainfall across the UK. Here the task of segmentation could go on endlessly, where one could use counties, parishes, constituencies, or council wards to segment the country and classify using their official political names. Instead I decided to avoid this by using the most obvious clusters and classifications;

1. Wales
2. Scotland
3. Northern Ireland
4. England

It has been assumed that the audience needs no clarification as to what and where constitutes each constituent nation. It must also be plainly stated here that the decision to refer to Wales and Scotland by their English names instead of their endonyms was made as the entirety of the rest of the NLG outputs are in English due to the creators unfortunate monolingualism and therefore consistency was chosen. In the event that either nation successfully petitions to have their endonym become their official name at the UN, this can be updated within the system. An issue with this segmentation is that the area of England accounts for little under twice the area of Scotland, and almost 10 times that of Northern Ireland. As before this ambiguity can affect interpretation and therefore the further segmentation of England is proposed thusly;

- 4.1. North England - { 53.4084N° - Scotland }
- 4.2. South England – { Rest of England }

Unlike most other decisions made, one cannot rely on discretion to determine where the boundary between North and South England lie, so I exercise my discretion to determine that North England constitutes all areas between Liverpool and Scotland, thus everywhere North of 53.4084N° that is not Scotland. It is out of scope of this project to determine a consensus or otherwise if this definition suffices. If the reader is so enraged at the prospect of an area being included or excluded in the North by this definition, the author suggests using the system created to determine whether the grass outside may be dry enough for them to touch. Latitude is use here to make determining North England easier in the course of programming later on.

Segmentation was further used to determine spatial quantifiers to describe the total coverage of areas that experienced rainfall. “It rained over Cardiff.” fail to answer how much of Cardiff experience rainfall as required by the system specification and as such the following segmentation is submitted;

1. All – {>95% coverage}
2. Most – { 56 – 94% coverage}
3. Half – {45 – 55% coverage}
4. Some – {26 – 44% coverage}
5. Parts – {<25% coverage}

Leeway has been built into the use of “All” and “Half” for separate reasons. For “All” it is simply that an area that has near total coverage would, in NL, be naturally described as “having seen rain over all of [it]”. For “Half”, some areas will be covered by an odd number of pixels and therefore discretion has been inbuilt as the data source unfairly restricts many regions from being described as “having seen rain over half of [it]”.

The final lexicalisation task determines how the system will refer to the time. While the aim of the system is to produce an NLD of rainfall, using the natural “o’clock” description of time seems outdated and overly complicates the extraction of temporal data from the radar data. As such it is determined that times will be presented in a digital clock format, i.e. “It rained between 9:00 and 10:00.” A 24 hour clock system is employed to avoid the necessity of using “am” and “pm” identifiers.

#### Referring Generation Expression.

This task was relatively brief in comparison to the one before as there are only 3 entities within the desired output that need to be considered and resolved. The first is the weather

itself, which in English always uses the singular third-person pronoun “it” and is therefore trivial. The second is the location of the rainfall, which for the sake of simplicity and clarity will always use the locations proper noun as its reference expression, and outputs will be structured to avoid the necessity of an additional referral expression, i.e. “It rained heavily in Glasgow.” as opposed to “It rained in Glasgow, it rained heavily there.” where “there” is used as an adverbial referring expression. Lastly, as previously alluded to, the time frame within which the rainfall is being considered must be contemplated. As with the location, the time frame can be reiterated however that generates a level of redundancy when the user is selecting which time frame to consider. As such, 2 determinations were made. When receiving an overview of the rainfall map of the whole UK, the actual time frame would be repeated back so as to not be forgotten in the larger amount of output data. When receiving a specific location for which to receive and NLD of rainfall data, the reference expression “that time” would instead be used to subtract some redundancy.

### Linguistic Realisation

The last task is mundane, with much of it having been achieved throughout the examples used to demonstrate potential outputs of the system in the previous tasks. The most trite decisions are as so;

- The capitalisation of proper nouns such as “Cardiff” or “Wales” is trivial, as is the use of capitals at the beginning of sentences.
- Outputs to be in the past tense are obvious as the data is neither current nor predicative, and the use of the simple past tense over the continuous is instinctual, though either could be used here accurately depending on one’s linguistic persuasion.
- While both prepositions “in” and “over” have been utilised in examples, and are used in everyday speech, the more “grammatically correct” one “over” will be adopted henceforth.
- After any determiners the preposition “of” must be employed. i.e. “It rained over all of Cardiff.”
- Prepositions “between” and “during” both have the same meaning in reference to time frames but are not interchangeable in all circumstances. Accordingly, “between” shall be used when two times are given, e.g. “between 9:00 and 10:00”, and “during” ought to be used when referring to a synoptic timeframe, e.g. “during that time”.



The only task here requiring any consideration here is how to apply the 8 adjectives to describe the rain. There are more solutions besides the one adopted, however modifying them into adverbs permitted for more concise and explicit outputs. “It rained heavily” rather than “The rain was heavy”. In this format, the order of “lightly moderate” and “firmly moderate” were switched so as to read “it rained moderately lightly” and “it rained moderately firmly” in order to make sense.

### Template

The above processes were undertaken so as to produce the template for which the system would output the source data into natural language. These templates are;

- “It rained [adverbial modifier] over [determiner] of [nation/region of England] between [n]:00 and [n+1]:00.”
- “It rained [adverbial modifier] over [determiner] of [specific location] during that time.”
- “It did not rain over [nation/region/location]”

Example outputs of what the system ought to produce are;

- “It rained lightly over most of Scotland between 8:00 and 9:00. It rained moderately over some of Northern Ireland. It rained heavily over half of Wales. It rained heavily over North England. It did not rain over South England.”
- “It rained heavily over most of Scotland and Northern Ireland and parts of North England between 9:00 and 10:00. It rained lightly over some of South England. It rained very heavily over all of Wales.”
- “It rained extremely heavily over Belfast during that time”

From these examples we note 2 things. Firstly, the time is only stated once during the overall description of the UK to avoid redundancy. Secondly, where locations experience the same level of rainfall, they system should aggregate them to avoid repetition, and an additional aggregation ought to occur if they also have the same coverage of rainfall.

Now the templates have been produced, the task of implementation can commence.

## Implementation

Having completed, through the methodology process, the content determination of both the NLG template and the system itself, we now move on to implementation. Here we look at how the task associated with the system were implemented with a Python framework.

The file directory containing the complete project should look like thus;

```
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\c1769632\OneDrive - Cardiff University\C1769632_Dissertation> dir

Directory: C:\Users\c1769632\OneDrive - Cardiff University\C1769632_Dissertation

Mode                LastWriteTime         Length Name
----                -
d----1           12/09/2022    13:45             RainImgs
-a---1           12/09/2022     00:05        613887 Eng.csv
-a---1           18/09/2022    14:09         2046 GetCountries.py
-a---1           02/08/2022    12:32          909 GetData.py
-a---1           18/09/2022    14:14        12036 London.csv
-a---1           18/09/2022    14:24         8147 NLGSystem.py
-a---1           11/09/2022    19:26       1027191 Scot.csv
-a---1           11/09/2022    14:26       248115 Wales.csv
-a---1           18/09/2022    13:59         131 workspace.code-workspace
```

*Figure 2 - Depicting the directory and its entries required for the system to run*

As aforementioned, the RainImgs folder contains the radar images of rainfall from the MET Office between the 6/12/21 and 17/12/21 provided by Dr Corcoran in PNG format. The code is designed to be able to retrieve from this file and others so long as the script file remains in the same folder as all other files. The GetData.py script includes some preliminary code used to download radar images from the past 24 hours. It is included in the file for completeness however ought not to be run, and if run will not work. The RainImgs file is used instead as a matter of practicality as attempting to utilise images taken during “the driest summer so far since 1976” is inopportune.

## Geocoding

The first programming task for the system was to create a way of referencing locations within the images that are not obvious to the eye. Referring back to Figure 1, it is not immediately obvious that the image is taken over the UK as there is no underlying map. The Met Office makes note that the image is taken between the coordinates 48° to 61° North and 12° West to 5° East and proffers that the image should be overlaid some map taken

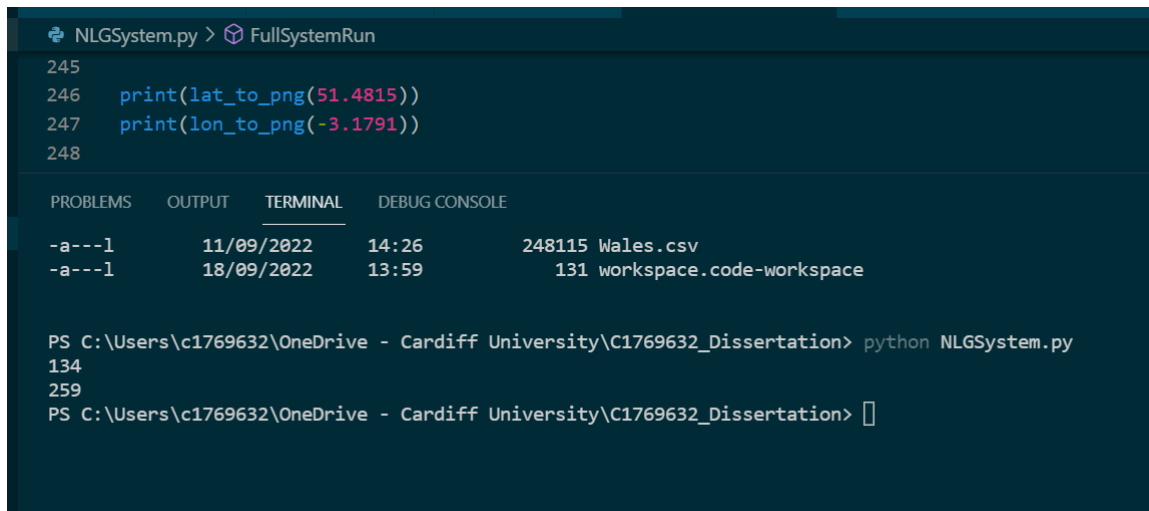
between those coordinates. Instead however, the system relies on basic mathematical transformations. The size of the PNG images is 500 x 500 pixels from [0,0] to [500,500], and the Map is 17 x 13 degrees from [-12,48] to [5,61]. Thus, all that is required to equate latitude and longitudinal coordinates to specific pixels on the map are horizontal and vertical translations and scaling. This leads to the following formula;

$$g(x) = \frac{500}{17} f\left((x + 12) \frac{500}{13}\right) - 48$$

Where the longitude coordinate is undergoing a horizontal shift 12 degrees to the left and being scaled up by a factor of 500/13 and the latitude coordinate is undergoing a vertical translation 48 degrees down and being scaled up by a factor of 500/17. This is performed in the code by 2 functions; `lat_to_png(lat)`, and `lon_to_png(lon)`, which take as arguments the latitude and longitude coordinates respectively.

“Why?”

Taking Cardiff to have latitude 51.4815 and longitude -3.1791 and inputting these figures into the respective functions, we are returned the following;



```

NLGSystem.py > FullSystemRun
245
246 print(lat_to_png(51.4815))
247 print(lon_to_png(-3.1791))
248

PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE
-a---1    11/09/2022  14:26    248115 Wales.csv
-a---1    18/09/2022  13:59    131 workspace.code-workspace

PS C:\Users\c1769632\OneDrive - Cardiff University\C1769632_Dissertation> python NLGSystem.py
134
259
PS C:\Users\c1769632\OneDrive - Cardiff University\C1769632_Dissertation> 

```

Figure 3 - Showing the system convert longitude and latitude to png coordinates

The latitude has been converted to a y coordinate and the longitude to an x coordinate. This means that Cardiff is located at pixel [259, 134] of the radar map png. The system can now take latitude and longitude data of a place and locate it in the radar image.

In reality, locations are neither single lat/long coordinates or single pixels on a map. Thus a bounding box of coordinates is required. 2 types of bounding box were utilised, both in the form:

Location = [Southern Most lat, Northern most lat, Western Most lon, Eastern Most, lon]

Firstly, bounding boxes for each constituent nation were created using their extreme points in all compass directions as follows ;

- Wales = [51.3667, 53.4333, -5.3, -2.65]
- Scotland = [54.6333, 59.5468, -13.6833, -0.7167]
- Northern Ireland = [54.0167, 55.3, -8.1775, -5.41667]
- England = [49.85, 55.8, -6.45, 1.7667]

All coordinates are taken from GeoHack (2022). As mentioned under Methodology – Lexicalisation, England is further segmented into North and South along the line of latitude 53.4084. It must be noted here that the latitude used as Scotland’s most northerly point is the northerly most point of Fair Isle and not that of the Shetland Islands. That is for the reason that the radar images do not contain data from the Shetland Islands so they are out of scope of the system.

Secondly, bounding boxes must be collected for and towns or cities a user may wish to input. This is done on an “as required” basis, demonstrated in a snippet of the code below;

```
165
166 def CityNLD(img): # gets NLD of rainfall on an inputed city or town level
167     city = input('Enter UK city or town:')
168     country = "UK"
169     url = "https://nominatim.openstreetmap.org/?addressdetails=1&q=" + city + "+" + country + "&format=json&limit=1"
170     response = requests.get(url).json()
171     boundary = response[0]["boundingbox"]
172     boundary_box = boundary[0:2]
173     if city.lower() == 'london':
174         city_tiles = (loadtxt("London.csv", delimiter=","),).astype(int)
175     else:
176         city_tiles = true_boundary(boundary_box, city)
```

Figure 4 - Retrieving boundary data for locations

When a user inputs their desired location, the system uses the `request.get(url)` function to retrieve data from OpenStreetMap.org, an open source site that proffers a wide range of geodata. Here is an example of the data retrieved when a user inputs “London”;

```
{
  "place_id": 339458995,
  "licence": "Data \u00a9 OpenStreetMap contributors, ODbL 1.0. https://osm.org/copyright",
  "osm_type": "relation",
  "osm_id": 65606,
  "boundingbox": [
    51.2867681,
    51.6918741,
    -0.5103751,
    0.3340155
  ],
  "lat": 51.5073219,
  "lon": -0.1276474,
  "display_name": "London, Greater London, England, United Kingdom",
  "class": "boundary",
  "type": "administrative",
  "importance": 0.9307827616237295,
  "icon": "https://nominatim.openstreetmap.org/ui/mapicons/poi_boundary_administrative.p.20.png",
  "address": {
    "city": "London",
    "state_district": "Greater London",
    "state": "England",
    "ISO3166-2-lvl4": "GB-ENG",
    "country": "United Kingdom",
    "country_code": "gb"
  }
}
```

#### OpenStreetMap (2022)

From this, we retrieve the “bounding box” data as demonstrated by the code.

Once the bounding boxes have been determined, the lat/long coordinates are converted into PNG coordinates. Before any image processing can take place however, each

pixel point within the bounding point must be reverse geo-referenced. This is because these bounding boxes are exceptionally large and will overlay other areas not in their remit. As can be seen with the country bounding boxes, Wales's boundary is contained entirely within that of England, as England's extreme points exceed Wales' in all direction. The `true_boundary(tuple,city)` function achieves this by reverting the png coordinates back to lat/long, reverse geo-referencing these latitudes with OpenStreetMap, at the following site;

```
"https://nominatim.openstreetmap.org/reverse?format=xml&lat="+str(lat)+"&lon="+str(lon)+"&zoom=18&addressdetails=1"
```

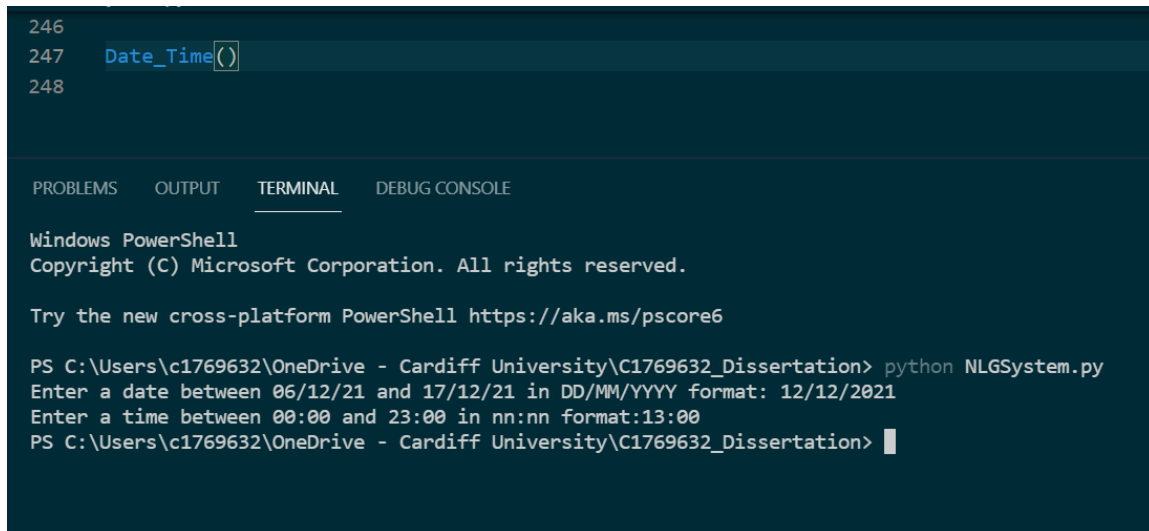
The function checks that the name of the city/country we want the boundary for is found within the data it retrieves, then appends to a list all those png coordinates whose check return a positive result. This turned out to be an incredibly time consuming task for the system. As such, England, Scotland, Wales, and London underwent reverse geo-referencing ahead of time using the `GetCountries.py` script. This script utilised the `true_boundary()` and its other associated functions ahead of time, and used numpy's `savetxt` function to save each point identified as accurately existing within each region to a csv file, saved as "Country.csv" and "City.csv". The script can be tested but it is recommended to use a small city as a test case. (The script took 8 hours to create the Eng.csv file). All other cities/towns remain to be retrieved on an "as required" basis. Northern Ireland's boundary cannot be reverse geo-referenced as OpenStreetMap does not differentiate between Northern Ireland and Ireland, and as such its bounding box is used as is. As there is minimal overlay between NI and Scotland's boundary box, and Scotland is reverse geo-referenced, this was determined to be acceptable.

It is the above functionality of the system that permits it successfully fulfil the first technical requirement of the NLG system of assisting the user in locating rainfall in areas of the image when an underlying map isn't present.

### Chrono-Requesting

The second technical task the system had to perform was taking as input the date and time that a user wanted to receive rain data for and retrieving the relevant images before returning the time within the NLG template already outlined.

The `Date_Time()` function is used to request the desired temporal data from the user. As demonstrated below, a user will first be asked to enter a date between the 6/12/21 and 17/12/21 in DD/MM/YY format. Then the user will be asked for a time between 00:00 and 23:00 for all dates except the 6<sup>th</sup> and the 17<sup>th</sup>. This is because for these dates no radar images are available before 11am for the former, nor after 7am for the latter, hence these are reflected in the input messages a user will receive.



```

246
247  Date_Time()
248

PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\c1769632\OneDrive - Cardiff University\C1769632_Dissertation> python NLGSystem.py
Enter a date between 06/12/21 and 17/12/21 in DD/MM/YYYY format: 12/12/2021
Enter a time between 00:00 and 23:00 in nn:nn format: 13:00
PS C:\Users\c1769632\OneDrive - Cardiff University\C1769632_Dissertation>

```

Figure 5 - Showing users input options

This function is then utilised in the `FullSystemRun()` function, which is the complete NLG system. The function extracts from the `Date_Time()` function the day and the hour for which the user requests a NLD of the rainfall data. It will then check to see if each quarter hourly radar image is available for that day and time. At least one image from within that hour are always available. From the available images, that will number between 1 and 4, the system selects that which has the largest quantity of rainfall over land. This is so the system may offer the most cautious description of rainfall at the given time. This image is then used as the input for the remaining two functions, `CountryNLD(rain_image, time)` and `CityNLD(rain_image)`, which as their names ought to suggest, produce the NLD of the rainfall radar data that fulfils the final technical requirement of the system.

### Precipitation Characterisation

There are minimal differences between the functions that return a NLD for rain over a specific City/Town/Other geographic location and the overarching summary of rainfall over the 3 country and 2 regions. Firstly, it is noted that `CountryNLD` takes the argument time where `CityNLD` does not. This is in keeping with the template generated under Methodology

where only the overall summary on a country basis of rain would publish a given time. The second is that the CityNLD makes use of the request function to access OpenStreetMap to choose specific locations to receive rainfall data on as illustrated under geo-coding. Otherwise, both function utilises the same sub-functions to generate their NLDs.

The `count_rain(tuple, img)` sub-function performs the task of image processing to determine which adverbs and determiners as outlined in lexicalisation ought to be admitted to be used by the `C___NLD()` functions in their templates to offer a textual summary of the visual data inputted. The two arguments `count_rain()` takes are the image file containing the png file to be analysed, and a tuple. This tuple is all the points within the image that need to be analysed, as determined using the `lat/long_to_png()` and `true_boundary()` functions.

Under lexicalisation we noted the 8 precipitation rates and their associated colours that the radar images employ. Their respective RGBA values were identified by inspecting the Met Offices UK Weather Map (2022) page and converting the respective Hex codes to RGBA. This permitted a count to be conducted to toll the occurrence of each individual colour. This count was used to determine which determiner and which adverb ought to be used in the NLD as thus;

- For the determiner, the total number of non-black pixels, (where an RGBA of [0,0,0,0] represents no rainfall) was divided by the total number of pixels constituting the area of the location being analysed. i.e. the number of entries in the tuple. This figure was then rounded to 2 significant figures, and as determined under lexicalisation, was mapped to the determiner whose range it fell within.
- For the adverb, the most recurrent rate of precipitation was taken and mapped to its associated adverbial modifier. This permits the system to give the most well rounded description of the overall situation in the desired location it can.

The `count_rain()` function then returns these modifiers into the NLD functions. The `CityNLD()` function receives these after having already taken user input for a specific location. If the determiner was mapped to “none”, then the system returns “It did not rain over (location)”, else the system will supplement the modifiers into the template “It rained (adverb) over (determiner) of (location)” as per requirement. The `CountryNLD()` takes the returned modifiers and populates the required template 5 times for each individual nation and the English regions. Thus the system is able to produce a desired NLD of rainfall from radar

images for a specific data and time for a range of locations as demonstrated in the next chapter.



### Evaluation and Analysis

The assessment of the end system is broken into 2 elements. Firstly, it shall be reviewed against test cases informed by the system requirements laid out previously, and then the output message will be assessed against the criteria adopted from Coch (1996).

The following 2 test cases are utilised to evaluate the two user requirements that required user input.

<b>Test Case Id: 1</b>	<b>Test Purpose:</b> Verify that Users can Input a date and time to choose a period to receive an NLD of.		
<b>Environment:</b> Using Google Chrome, running under Windows 10			
<b>Preconditions:</b> All files are in same directory			
<b>Test Case Steps:</b> These steps should be repeated thrice, firstly for date 6/12/21, then 17/12/21, then any date between the two			
Step No	Procedure	Expected Response	Pass/Fail
1	In the command terminal, the script NLGSystem.py should be ran.	User should see a string that states “Enter a date between 06/12/21 and 17/12/21 in DD/MM/YY format:”	Pass
2	User should input a date. Test data <06/12/21>	User should see a string that states “Enter a time between 11:00 and 23:00 in nn:nn format:”	Pass
3	User should input a time. Test data <17:00>	User should see a NLD of rainfall from that date.	Pass
4	User should input a date. Test data <12/12/21>	User should see a string that states “Enter a time between 00:00 and 23:00 in nn:nn format:”	Pass
5	User should input a time. Test data <13:00>	User should see a NLD of rainfall from that date.	Pass
6	User should input a date. Test data <17/12/21>	User should see a string that states “Enter a time between 00:00 and 07:00 in nn:nn format:”	Pass
7	User should input a time. Test data <06:00>	User should see a NLD of rainfall from that date.	Pass
<b>Related Tests:</b> Test Case 2			
<b>Author:</b> 1769632		<b>Checker:</b>	

Test Case 1

<b>Test Case Id: 2</b>	<b>Test Purpose:</b> Verify that Users can Input a location to choose a period to receive an NLD of.		
<b>Environment:</b> Using Google Chrome, running under Windows 10			
<b>Preconditions:</b> The previous test cases have been run.			
<b>Test Case Steps:</b> These steps should be repeated in conjunction with the steps in Test Case 1			
Step No	Procedure	Expected Response	Pass/Fail
1	In the command terminal, the script NLGSystem.py should be run.	User should see a string that states “Enter a date between 06/12/21 and 17/12/21 in DD/MM/YY format:”	Pass
2	User should have followed steps 2 & 3 , 4 & 5, or 6 & 7	User should see an NLD of rainfall from the associated dates. User should see the string “Enter UK city or town:”	Pass
3	User should input a location. Test data <London>	User should see a NLD of rainfall from that date.	Pass
4	User should input a location. Test data <Edinburgh>	User should see a NLD of rainfall from that date.	Pass
5	User should input a location. Test data <Cardiff>	User should see a NLD of rainfall from that date.	Pass
<b>Related Tests:</b> Test Case 1			
<b>Author:</b> 1769632		<b>Checker:</b>	

## Test Case 2

Below the associated responses from the test cases are demonstrated.

```

GetCountries.py  London.csv  GetData.py  NLGSystem.py X
NLGSystem.py > ...
244
245 FullSystemRun()

PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\c1769632\OneDrive - Cardiff University\C1769632_Dissertation> python NLGSystem.py
Enter a date between 06/12/21 and 17/12/21 in DD/MM/YY format: 06/12/21
Enter a time between 11:00 and 23:00 in nn:nn format:17:00
It rained slightly over parts of Scotland during that time.
It rained moderately over some of Wales during that time. It did not rain over Northern Ireland. It rained moderately over parts of
South England during that time.
Enter UK city or town:London
It rained slightly over parts of London during that time.
PS C:\Users\c1769632\OneDrive - Cardiff University\C1769632_Dissertation> python NLGSystem.py
Enter a date between 06/12/21 and 17/12/21 in DD/MM/YY format: 12/12/21
Enter a time between 00:00 and 23:00 in nn:nn format:13:00
It rained slightly over parts of North England between 13:00 and 14:00. It rained slightly over parts of Scotland during that time. I
t rained slightly over parts of Wales during that time. It did not rain over Northern Ireland. It did not rain over South England.
Enter UK city or town:Edinburgh
It did not rain over Edinburgh
PS C:\Users\c1769632\OneDrive - Cardiff University\C1769632_Dissertation> python NLGSystem.py
Enter a date between 06/12/21 and 17/12/21 in DD/MM/YY format: 17/12/21
Enter a time between 00:00 and 07:00 in nn:nn format:06:00
It did not rain over North England between 06:00 and 7:00. It rained moderately lightly over none of Scotland during that time. It di
d not rain over Wales. It did not rain over Northern Ireland. It did not rain over South England.
Enter UK city or town:Cardiff
It did not rain over Cardiff

```

Figure 6 - Showing full system output

As evidenced by the success of the system in passing the test cases, and the associated output, we can evaluate that it has succeeded in achieving the goals of all the laid out design specifications.

Now the NLD output will be assessed against Coch's criterion as highlighted in the literature review. Each criterion will be assessed out of 10 based on the output.

- Correct spelling – [8/10]

While the output itself has impeccable spelling by construction, marks were deducted here as if the same is not true of the user, the system will fail to execute. If a user misspells a location, the system will not be able to recall from OpenStreetMap any spatial and will fail. This is demonstrated in the figure below where "Cardiff" has been misspelled as "Carfid"

- Good grammar – [10/10]

While "Good Grammar" can be hard to quantify in a language system that doesn't have formalised grammar rules akin to the "Academie Francaise", we can say that, by construction, the grammar of the output is correct and cannot deviate in accuracy so this is a trivial check.

- Comprehensiveness – [6/10]

Comprehensiveness suffers through the construction of the template, rather than the success of the programme itself. Whilst the system breaks down the rainfall at a country and regional level, these are constructed superficially and could be further broken down to offer a more detailed and exhaustive account. While the user is able to input their desired locations for a look at a more unique locale, perhaps a system that pre-prepares analysis of more microscopic areas such as at a council or constituency level, would return a higher rate of comprehensiveness.

- Rhythm and Flow – [5/10]

While the individual sentences have been designed to have a natural rhythm and flow, the output on the whole loses marks here as a consequence of lack of connectivity between sentences, and lack of aggregation of similar clauses that leads to repetition.

- Appropriateness of the tone – [7/10]

Appropriateness of tone here loses marks as the system cannot adapt the output to different users and relies on a "universal audience". If the system were to be utilised by a child, or

perhaps someone in denial of changes to weather patterns, the system would benefit from being able to adjust output accordingly to a more child-friendly tone. This extends to any audience who may not appreciate the rigidity of the output.

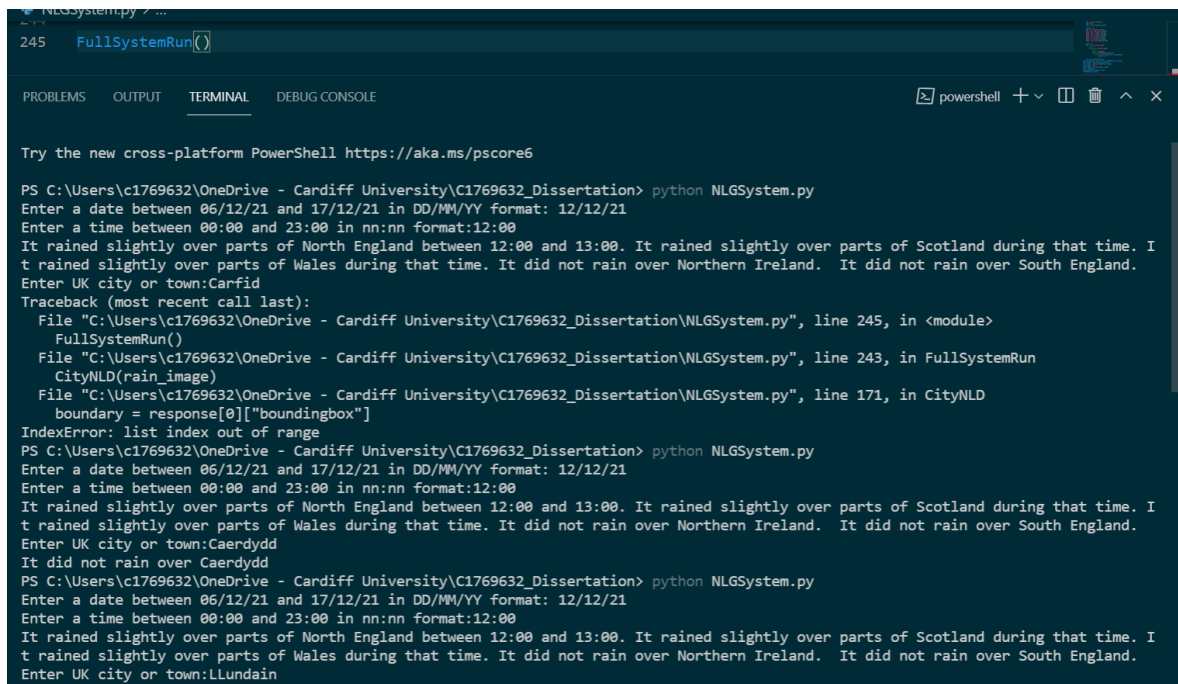
- Absence of repetition – [4/10]

While individual sentences are free from repetition, and the system succeeded in abstaining from repeating the time for which rain events occurred, it did not succeed in removing the repetition of the same level of rainfall experienced across countries and regions. The result is an accurate but disarming NLD of rainfall across the UK.

- Correct choice and precision of the terminology used – [8/10]

As the system uses a template formula, the evaluation of the correct terminology must be done against the justification made for the linguistic mapping in the Lexicalisation section of the Methodology chapter. While the Met Office's guide to precipitation rates is a professional standard, it doesn't take into account cultural differences in how people experience rainfall. Those in London may have a different interpretation of what constitutes light rainfall than those used to a wetter Welsh environment and as such the system risks either downplaying the severity of the rain in one location or overstating it in another. This thus bleeds into the comprehensiveness criterion also. In the Discussion chapter we shall look at possible solutions to this issue but for now it suffices that the system does accurately map the desired quantities to the relative linguistic syntax as pre-determined through the Methodology.

Despite these failings, the system does succeed the original aims laid out in removing a user's need to have any of the pre-determined technical insights to interpret the radar data at their disposal. In the next chapter, possible solutions to these issues will be looked at in conjunction with how changes to the system could also increase its prospects in future applications.



```
PS C:\Users\c1769632\OneDrive - Cardiff University\C1769632_Dissertation> python NLGSystem.py
Enter a date between 06/12/21 and 17/12/21 in DD/MM/YY format: 12/12/21
Enter a time between 00:00 and 23:00 in nn:nn format:12:00
It rained slightly over parts of North England between 12:00 and 13:00. It rained slightly over parts of Scotland during that time. I
t rained slightly over parts of Wales during that time. It did not rain over Northern Ireland. It did not rain over South England.
Enter UK city or town:Carfid
Traceback (most recent call last):
  File "C:\Users\c1769632\OneDrive - Cardiff University\C1769632_Dissertation\NLGSystem.py", line 245, in <module>
    FullSystemRun()
  File "C:\Users\c1769632\OneDrive - Cardiff University\C1769632_Dissertation\NLGSystem.py", line 243, in FullSystemRun
    CityNLD(rain_image)
  File "C:\Users\c1769632\OneDrive - Cardiff University\C1769632_Dissertation\NLGSystem.py", line 171, in CityNLD
    boundary = response[0]["boundingbox"]
IndexError: list index out of range
PS C:\Users\c1769632\OneDrive - Cardiff University\C1769632_Dissertation> python NLGSystem.py
Enter a date between 06/12/21 and 17/12/21 in DD/MM/YY format: 12/12/21
Enter a time between 00:00 and 23:00 in nn:nn format:12:00
It rained slightly over parts of North England between 12:00 and 13:00. It rained slightly over parts of Scotland during that time. I
t rained slightly over parts of Wales during that time. It did not rain over Northern Ireland. It did not rain over South England.
Enter UK city or town:Caerdydd
It did not rain over Caerdydd
PS C:\Users\c1769632\OneDrive - Cardiff University\C1769632_Dissertation> python NLGSystem.py
Enter a date between 06/12/21 and 17/12/21 in DD/MM/YY format: 12/12/21
Enter a time between 00:00 and 23:00 in nn:nn format:12:00
It rained slightly over parts of North England between 12:00 and 13:00. It rained slightly over parts of Scotland during that time. I
t rained slightly over parts of Wales during that time. It did not rain over Northern Ireland. It did not rain over South England.
Enter UK city or town:LLundain
```

Figure 7 - Demonstrating faults in the system in line with the above criterion marking

## **Discussion**

The system is successful in meeting the aims laid out previously. That is to say, given a time series of radar maps depicting rainfall data, the system should output in natural language a spatial description of where the rain fell, a temporal description of when the rain fell, and describe how heavy the rainfall was and the percentage of the area it covered in natural language. We now must consider the limitations and short comings of the system, and how amendments to the system can also widen its scope and utility.

As noted in the evaluation section, the system cannot identify spelling errors in location names submitted by users. This not only affects the effectiveness of the system but also its accessibility, as any user who suffers from word processing issues may not be able to receive the input they desire. Ways in which accessibility for these issues can be assured, beyond standard spell checking and autocorrection tools, include;

- Integrating a visual map, for example in D3, that removes the need to type in a location and permits a user to select the location, or a general area, for which they wish to receive rain data for.
- Integrating the system with audio receiving devices such as Amazons “Alexa” to allow users to search for places orally. Many voice devices already permit users to search for weather forecasts in this way, but our system could be integrated to allow users to request the rain data for previous dates as is its function. Not only would this assist those with written issues, but also make it more accessible to those with visual impairments.

Continuing on from the second suggestion above, while the next most obvious development is having the system capable of text to speech descriptions, the basic sentence structure and simplicity of sentences means that the system could be integrated with braille technology to produce descriptions accessible to those with severe visual impairments. The Tactile text-to-braille machine reported upon by Mashable (2017) would be an ideal candidate for testing such integration and expanding the systems accessibility.

Special care would need to be taken if the system were to produce braille text, in that any text outputted would need to respect the languages own distinct grammatical rules. The same is true for any system that seeks to generate output in any language other than a Central Language (CL). Streiter et al (2006) identified CLs to be English, French, German, Russian,

Mandarin, and Japanese, languages that receive a greater volume of interest and development in the field of NLP. It must be argued that braille would constitute a Non-Central Languages (NCLs) as would Welsh, Gaelic, Gaeilge, Manx, and Kernow. While the system is able to receive and retrieve Welsh language place names in the data, this is merely a “happy accident”, thanks to the extensiveness of OpenStreetMaps open data. I am unable to say definitively for how much of the UK the system is capable of doing this for. However this does afford the system a great opportunity in being developed further to operate in these languages and be increasingly more accessible to a variety of users across the UK. Common practice for NLP systems that output NCLs is to simply first operate in a CL then translate the output accordingly. This does NCLs a disservice as the peculiarities and nuances of these more unique languages cannot be conveyed in this way and are lost. The dominance of English online and in the digital world could be argued to be contributing to the decline of many more marginalised languages, and so the opportunity to produce NCL text through systems like this one should not be over looked in correcting this. While Welsh, Gaelic, and Gaeilge still benefit from having a somewhat robust speakership, the extent to which this is shared differs, and is no where near matched by Kernow and Breizh. In the same way the simplicity of the sentence structure benefits braille, the same is true to all 5 of these languages, and adapting the system to produce output in these languages from scratch instead of translating, and geo-pinning such outputs to their respective locations, can improve the accessibility to non-English language tools across the UK. It must be noted that such a task would require either to collect a new database of locations with the respective Celtic names, or to translate location names from English to one of the above and vice versa, however developments on such a project could go ways into helping such targets as the Welsh Senedd’s “One Million Welsh speakers by 2050” and similar.

Following on from the two above, the comprehensiveness of the system could be improved by offering regionalised descriptions that take into account cultural attitudes to rain. Further investigation with ethnographic dynamics would need to be conducted but preliminary research in the form of questionnaires could produce specialised messaging that makes use of GPS and location services already integrated into most digital appliances to better convey rainfall severity in different areas. A potential hazard in this is that egotistical bravado in the sampling could result in ineffective descriptions that downplay the severity of weather and impact users understanding of the consequences of such precipitation events. While large sampling sets can be utilised to circumvent this, more technical understandings

on the impact of rain could be consulted to adapt the mapping of the precipitation rate to an adequate adverbial modifier. For example, using engineering research into the impact of rain on buildings could be used to offer more detailed NLDs that can convey the consequences of a precipitation rate on structures to help communicate more aptly the level of rainfall. Indeed, such a system could be used to convey in natural language details to architects, engineers, and builders details on how rainfall might impact not only on the time line of a build or development, but on the integrity of it thereafter and could be used as a warning system.

In the introduction the scope of this project was clearly defined and the system was limited to dealing singularly with radar maps detailing rainfall across the UK. This scope can of course be widened to utilising radar maps from across the world, and this could be used in conjunction with projects on NCLs to produce textual description of rain in many countries. Rain however is not the only weather event on offer from the MET Offices DataPoint service. Radar data pertaining to lightning strikes and surface pressure charts could be integrated into the system to produce more well-rounded NLD weather observations. As alluded to, a system that can translate into natural language the increased frequency of weather events can help convey the consequences of changing weather patterns caused by climate change. Instead of using complicated technical representations of data, the simple and digestible sentences demonstrated could better convey how changes in frequency, and range between, weather extremes is not consistent with regular weather patterns. Furthermore, integrating a wider range of weather events can expand upon the initial scope of this project which omitted describing the cause of the rain events to instead include them, as the inclusion of pressure and temperature spatio-temporal data can be included in mapping to produce text that could be of the form – “It rained heavily in Cardiff between 9:00 and 10:00, following a period of high pressure over the Irish sea between 0:00 and 5:00 and high temperatures over the Channel.” Please note that this is an example of what could be outputted and would require further work to ensure that the information and understanding is metrologically accurate. The inclusion of further data can not only widen the projects scope and applicability but also improve upon the comprehensiveness of the systems output.

Some may question the usefulness of a system that returns data for historic rain events, and if they cannot be persuaded by some of the aforementioned usage suggestions above, then perhaps the future applicability below can suffice. Systems already exist that can convert to NL weather forecast data, but much of these systems rely on statistical meteorological processes to extract the data from. Nowcasting is a form of short-term



weather forecasting that provides weather forecasts no more than for 2 hours ahead based on current weather observations. DeepMind have developed software that produces these short term forecasts for rainfall that generates radar map “movies” that portray the potential rainfall observations over the next 2 hours. As the system already takes radar images as input, the adaptation of the system to these generated weather models could be used to produce NL forecasts of the imminent predicted weather that is typically more accurate than traditional day ahead forecasting.

Whilst the scope of the project limited the end system that was produced, it is clear that the task of widening that scope can be readily achieved, and what has been produced in the course of this project is more of a starting product than an end product, and can be nurtured further, from a single raindrop, to a flood of opportunity.

## **Conclusion**

From the outset of this project, the goal has been to produce a system capable of rendering radar images of rain observation in simple human language descriptions to convey the data within to a wide ranging audience. While that has been achieved, it is not the soul outcome of this project. Now we will conclude this report with an overview of all that has been learnt and achieved.

First, we surmise that the field of NLG is long established but ever evolving to encompass the traditional formats and procedures used to develop texts with modern techniques that allow computer to act as linguists in choosing syntax and lexis to convey the information that is requested. But in building this system, we rely on the tried and tested methods to generate an accurate and astute description of rainfall as desired.

As much as we may wish, the process of programming cannot be separated from more “administrative” tasks, and the neither should it be. In following a regiment of system design and specification we can ensure that the aims and objectives of the project were kept in the forefront of our minds to ensure the end product didn’t fall short of the minimum expectations to suffice the requirements defined.

Where Computing and Mathematics as STEM subjects may be viewed as the antithesis of English language and the arts, we see that the two are conjoined intrinsically in this project, and without an understanding of both the end goal could never have been achieved. In the course of pursuing Reiter and Dales 6 NLG tasks, the exhaustive linguistic hurdles that had to be cleared truly entrenched the importance of cross-disciplinary collaboration in any academic pursuit.

In turning the theoretical into the practical, the employment of mathematical principles to manipulate differences in data into agreement shows the simplest of solutions can solve those tasks that at first seem tremendously difficult, and mathematics is the truest footholds we can rely on.

In assessing the overall product, we see that perfection is impossible, but so it should be. For failure affords opportunity and the ideas explored in finding solutions to the issues that arose gave rise to new avenues of exploration that ensure the projects of yesterday can and will be picked up off the shelf tomorrow and be taken forward in new lights to reach new heights.

## References

1. Reiter, E. and Dale, R., 1997. Building applied natural language generation systems. *Natural Language Engineering*, 3(1), pp.57-87.
2. Dale, R., 2020. Natural language generation: The commercial state of the art in 2020. *Natural Language Engineering*, 26(4), pp.481-487.
3. Coch, J., 1996. Evaluating and comparing three text-production techniques. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*.
4. Goldberg, E., Driedger, N. and Kittredge, R.I., 1994. Using natural-language processing to produce weather forecasts. *IEEE Expert*, 9(2), pp.45-53
5. Reiter, E., Sripada, S., Hunter, J., Yu, J. and Davy, I., 2005. Choosing words in computer-generated weather forecasts. *Artificial Intelligence*, 167(1-2), pp.137-169.
6. Jebson, S., 2007. National Meteorological Library and Archives-Fact sheet No. 3-Water in the atmosphere. MET Office, 3, pp.1-27.
7. Turner, R., Sripada, S., Reiter, E. and Davy, I.P., 2006. Generating spatio-temporal descriptions in pollen forecasts. In *Demonstrations* (pp. 163-166).
8. Turner, R., Sripada, S., Reiter, E. and Davy, I.P., 2008, May. Building a parallel spatio-temporal data-text corpus for summary generation. In *The Workshop Programme Methodologies and Resources for Processing Spatial Language* (p. 28).
9. LaPlante, P.A., 2017. *Requirements engineering for software and systems*. Auerbach Publications.
10. Sommerville, I., 2005. Integrated requirements engineering: A tutorial. *IEEE software*, 22(1), pp.16-23.
11. Cohn, M., 2004. *User stories applied: For agile software development*. Addison-Wesley Professional.
12. GeoHack (2022). GeoHack – List of Extreme Points of England (Substitute England accordingly). Available at [https://geohack.toolforge.org/geohack.php?pagename=List\\_of\\_extreme\\_points\\_of\\_England&params=52\\_29\\_N\\_1\\_46\\_E](https://geohack.toolforge.org/geohack.php?pagename=List_of_extreme_points_of_England&params=52_29_N_1_46_E) (Accessed: 7 August 2022)
13. OpenStreetMap(2022) About Open Street Map. <https://www.openstreetmap.org/about> (Accessed : 10 August 2022)
14. OpenStreetMap(2022). <https://nominatim.openstreetmap.org/?addressdetails=1&q=LONDON+UK&format=json&limit=1> (Accessed : 12 August 2022)
15. Mashable (2017) How six young women invented a life-changing device in less than a day <https://mashable.com/article/team-tactile-braille-display> (Accessed : 19 September 2022)

16. Met Office (2022) UK Weather Map  
<https://www.metoffice.gov.uk/public/weather/forecast/map/#?map=Rainfall&fcTime=1663473600&zoom=5&lon=-4.00&lat=55.45> (Accessed : 3 August 2022)
17. DeepMind (2022) Nowcasting the next hour of rain <https://deepmind.com/blog/nowcasting-the-next-hour-of-rain> (Accessed : 2 September 2022)