Radboud University

Faculty of Science

BuienradarGPT

Leveraging LLMs for generating Local Weather Reports

Thesis MSc Computing Science

Specialization: Data Science

Supervisor:
Dr. Arjen P. de Vries

Author:
Megha Soni
(S1100529)

External Supervisor:
Dr. Daan Odijk @ RTL

Second reader:
Dr. Faegheh Hasibi

July 2024

Abstract

Accurate and timely weather information is crucial for planning daily activities and ensuring safety, in the Netherlands. Traditional weather reports usually provide forecasts for the entire country, which may not adequately address local variations. Large language models are renowned for their ability to generate coherent text. However, they are limited by the data they were trained on and can sometimes produce inaccurate or "hallucinated" information. To overcome these challenges, we introduce BuienradarGPT, an innovative pipeline that generates detailed local weather reports using Retrieval Augmented Generation (RAG). The generated reports are evaluated by human reviewers to assess their acceptance and the confidence users place in AI-generated weather reports. Our findings reveal that while AI-generated weather reports are well-received, there is room for improvement in making the reports more human-like and satisfying to a broader audience. Furthermore, we have developed a weather assistant chatbot that allows users to engage with weather forecasts, offering a more accessible and interactive method for obtaining weather information.

Contents

Chapter 1

Introduction

1.1 Related Work

Weather reports play a crucial role in the daily lives of people by aiding them in making informed decisions to protect their lives, property, and well-being [20, 19]. These reports are essential not only for severe weather situations but also for everyday activities. They help individuals and professionals alike to plan their activities efficiently and avoid potential risks and costs associated with adverse weather conditions. According to a study by Doksaeter Sivle and Kolsto [8], weather-related decision-making processes are significantly influenced by the amount of information provided in weather reports. People use varying amounts of information based on the importance of their activities and the suitability of the weather conditions.

While Buienradar [2], a platform providing current radar and weather information for the Netherlands and Europe, offers comprehensive weather reports for the entire country. However, presenting localized weather reports to users could enable more informed decisions, especially since weather conditions can vary across different geographical locations within the Netherlands.

This demand for localized and detailed weather information is well-supported by the advancements in AI and machine learning, which are continually improving the accuracy and relevance of weather forecasting. The survey by Chen et al. provides a comprehensive overview of AI methodologies specifically engineered for weather and climate data, highlighting the crucial role of foundation models in this field [4]. Deep learning models have shown remarkable promise in understanding and predicting weather patterns by processing large volumes of spatio-temporal and textual data. These models excel at extracting intricate patterns and relationships from data, which are essential for weather forecasting and comprehensive climate analysis. Various applications, such as forecasting, climate text analysis, and extreme weather prediction, have benefited significantly from these deep learning techniques.

Despite the success of deep learning models, the use of large language models (LLMs) in weather and climate forecasting is relatively new and still in its early stages. The recent advancements in AI and machine learning have driven interest in adapting LLMs, traditionally used in natural language processing (NLP), for weather and climate-related tasks. This emerging field leverages the versatility and advanced capabilities of LLMs to handle complex and diverse datasets.

For example, OceanGPT [3] represents the significance of LLM in the domain of ocean science, crucial due to the vast and complex nature of ocean data. Similarly, ClimateBERT [24] has been developed to tackle the limitations of general LLMs in processing climate-related texts. General language models often struggle with the specific terminology and context found in climate literature, which can limit their effectiveness. ClimateBERT, a transformer-based model further pre-trained on over 2 million paragraphs of climate-related texts from diverse sources, significantly improves the accuracy of text classification, sentiment analysis, and fact-checking tasks. This results in a substantial improvement in handling climate-specific language, showcasing its superiority for these tasks.

These advancements underscore the potential of LLMs in weather and climate research. We can leverage these models to analyze and interpret local weather data to generate detailed local weather reports based on user queries, and also enable interaction with a Weather Assistant.

However, LLMs come with several inherent challenges, such as unpredictability in responses and reliance on static training data, which can result in outdated information. For instance, if an LLM was trained using data up to 2020, it would not be aware of any weather patterns or significant meteorological events that occurred after that date. Key challenges associated with LLMs in weather forecasting include [17]:

1. Presenting false information when the model does not know the answer.

2. Providing outdated or generic weather data when users expect specific, current forecasts.

3. Generating weather predictions based on non-reliable sources.

4. Creating inaccurate weather reports due to terminology confusion, where different data sources might use the same terms differently.

In weather-based applications, these issues can impact users' trust, making it undesirable for the applications to rely solely on static data. One method to enhance the model's knowledge is by fine-tuning it on the latest weather data. However, this approach has several limitations [17]. Firstly, it can be costly and time-consuming. Secondly, the model's parameters may not be sufficient to learn all the necessary meteorological knowledge. Lastly, fine-tuning is not additive and may overwrite existing knowledge with new information. For example, a language model extensively fine-tuned on current weather data may lose its understanding of historical weather patterns. These issues highlight the need for more efficient and reliable methods to update and maintain the accuracy of weather-based applications.

1.2 Research Questions

The central focus of my Master's thesis lies in the application of large language models to analyze and interpret local weather data to generate comprehensive and informative weather reports.

At present, the weather reports generated by Buienradar are based on various satellite images and the data derived from the central weather station located in De Bilt. These reports are focused on providing a generalized overview of climatic conditions for the entire Netherlands. It requires manual composition by the meteorologists, a process repeated 3-4 times daily, involving synthesizing data from various satellite images and predicted variables. The labour intensive nature of writing these reports for every location in the country underscore the need for an alternative approach. Hence, we want to leverage large language models to generate insightful weather reports for different locations in the country. The focal points of inquiry in this study are as follows:

RQ1 What are the expectations of users visiting the Buienradar website within the Netherlands regarding the daily weather insights for specific locations?

We aim to understand the key preferences and requirements of the users accessing the Buienradar platform. Specifically, we want to identify the key weather parameters that hold significance for individuals. Remainder thesis ensures the inclusion of these aspects in the generated weather reports.

RQ2 How do individuals perceive weather reports generated by Artificial Intelligence (AI) systems?

By exploring users' attitudes, beliefs, and opinions regarding the reliability, accuracy, and overall quality of AI-generated forecasts, we seek to gain insight into the level of acceptance and confidence users place in AI-generated weather reports.

RQ3 Can the Retrieval Augmented Generation (RAG) approach provide a personalized weather chat experience for users?

We aim to explore whether RAG methodology used for generating weather reports can be adapted to develop an interactive chat feature that addresses individual users' specific weather-related queries and preferences.

Chapter 2

Background

2.1 Retrieval Augmented Generation (RAG)

In the attempt of enhancing the performance of LLMs, unsupervised fine-tuning has shown some potential. However, recent studies have demonstrated that Retrieval Augmented Generation consistently outperforms unsupervised fine-tuning [22]. Specifically, RAG not only excels in using existing knowledge encountered during training but also in integrating entirely new information. This is particularly relevant for applications that require domain specific knowledge, such as weather forecasting. One such tool is presented in the paper [5], which introduces FloodBrain. It uses web-based RAG with an LLM to generate detailed and accurate reports on flood events. These advancements underscore the value of RAG in enhancing the accuracy and reliability of LLM outputs in domain-specific contexts.

RAG is designed to address these challenges by introducing an information retrieval component that enhances the LLM's capabilities [7]. With RAG, the process of generating a response is augmented by first retrieving relevant information from new data sources based on the user input. This retrieved information, along with the user's query, is then provided to the LLM as context. The LLM uses this context data along with its pre-existing knowledge to generate appropriate and informed weather reports.

RAG technology offers several benefits to our weather forecasting project:

1. Cost-effective Implementation: RAG allows the introduction of new weather data to the LLM without the need for extensive retraining, making the integration of generative AI more accessible and practical.

2. Relevant Information: RAG ensures that users receive the most up-to-date weather information by connecting the LLM to live data sources, such as weather APIs, which provide real-time forecasts and meteorological data.

3. Enhanced User Trust: By enabling the LLM to provide weather forecasts with source attribution, RAG increases user trust and confidence. Outputs can include references to the original data sources for further verification and detail.

4. Developer Control: RAG provides developers with the ability to test and improve weather forecasting applications efficiently. They can modify the information sources to meet evolving requirements, restrict sensitive information retrieval based on authorization levels, and ensure appropriate response generation.

In our project, RAG plays a major role in generating accurate weather reports. RAG retrieves specific weather data based on user queries, which would include parameters such as location and date. This retrieved data is then utilized to produce detailed weather reports for the specified location and date, enhancing the overall relevance of the generated outputs. Essentially, RAG feeds the LLM with information from diverse sources, selected based on domain knowledge. By leveraging RAG, we aim to ensure that our AI-driven weather reports are based on factual, up-to-date information, providing valuable insights to users.

2.2 Llamaindex Framework

Following our exploration of RAG and its significant role in enhancing the performance of the LLMs, we turn our attention to the LlamaIndex [12] framework. This framework, previously known as GPT Index, is particularly well-suited for building context-augmented applications using the LLMs [18], and a central component in our weather report generation system.

LlamaIndex provides a comprehensive suite of tools for developing applications that leverage private or domain-specific data to enhance the capabilities of the LLMs. This is essential because, while the LLMs are trained on vast amounts of publicly available data, they often lack specific knowledge pertinent to certain domains, such as localized weather forecasting. By integrating RAG principles, LlamaIndex facilitates the ingestion, processing, and querying of private data efficiently, ensuring that the generated outputs are both relevant and accurate.

Overview

The primary function of LlamaIndex is to build context-augmented LLM applications, where context augmentation refers to the application of the LLMs on top of private or domain-specific data [12, 26]. This framework addresses the inherent limitations of the LLMs by providing tools that allow these models to access and utilize specialized datasets effectively.

- **Data Connectors:**
  LlamaIndex provides data connectors as the lowest layer of its architecture, which are designed to ingest data from a wide array of native sources and formats, including APIs, PDFs, and SQL databases. This capability unifies diverse data formats into a consistent structure that LLMs can easily process.

- **Data Indexes:**
  Once the data is ingested, LlamaIndex structures it into intermediate representations known as vector embeddings, optimized for LLM consumption. These vector embeddings are numerical representations that capture the meaning of the data, allowing the LLM to understand and retrieve relevant information based on semantics rather than simple keyword matching.

  Data indexes, LlamaIndex data structures to help in quickly retrieving relevant context for a user queries, form the foundation for RAG use-cases. Indexes are built from documents and used to build Query Engines to allow interactions over the data. Under the hood, the index stores its data in two formats: Node objects, which represent chunks of the data, and their corresponding embeddings in a Vector Store. By transforming the data into vector embeddings, LlamaIndex ensures that the information ingested into the architecture is accessible, and understandable by the LLMs.

- **Engines:**
  Query Engines, built on top of these indexes, enable LLM system to retrieve and integrate relevant information before generating a response. Retrieval involves finding and returning the most relevant data for a user query retrieving it from the vector store Index based on similarity between query and indexed information in embedding space, as well as metadata. It then combines the user query, the most relevant data as the context, and the prompt, which are together sent to the LLM to generate a response.

- **Observability and Evaluation:**
  To ensure the continuous improvement of applications, LlamaIndex integrates tools for rigorous experimentation, evaluation, and monitoring. This feature supports a virtuous cycle of development, where feedback and performance metrics drive ongoing enhancements.

Chapter 3

Methodology

In this section, we address the research question by generating AI-driven reports using the technology developed in this chapter. Our approach utilizes a sophisticated report generation pipeline, illustrated in Figure 3.1, which transforms raw CSV weather data into detailed and informative weather reports.

3.1 Report Generation Pipeline

The report generation pipeline begins by importing CSV files into Pandas DataFrame objects for initial preprocessing, a process detailed in Section 4.2, ensuring data quality and consistency.

Post preprocessing, the data is loaded into an in-memory SQLite database using SQLAlchemy, creating a structured SQL database. This database is then indexed (see Section 3.1.2) using LlamaIndex, which includes an object index for table schema and a vector store index for data representation, facilitating efficient querying.

The query engine from LlamaIndex, discussed in Section 3.1.4, converts natural language queries into SQL commands, retrieving the necessary data based on the user's query parameters: location and date. This ensures that the relevant weather data for the specified location and date is accessed. OpenAI's language model then synthesizes this data into comprehensive weather reports, structured by analyzing different times of the day and combining these sections.

This pipeline ensures data processing, relevant retrieval, and report generation, providing detailed weather reports based on integrated data from multiple csv files.

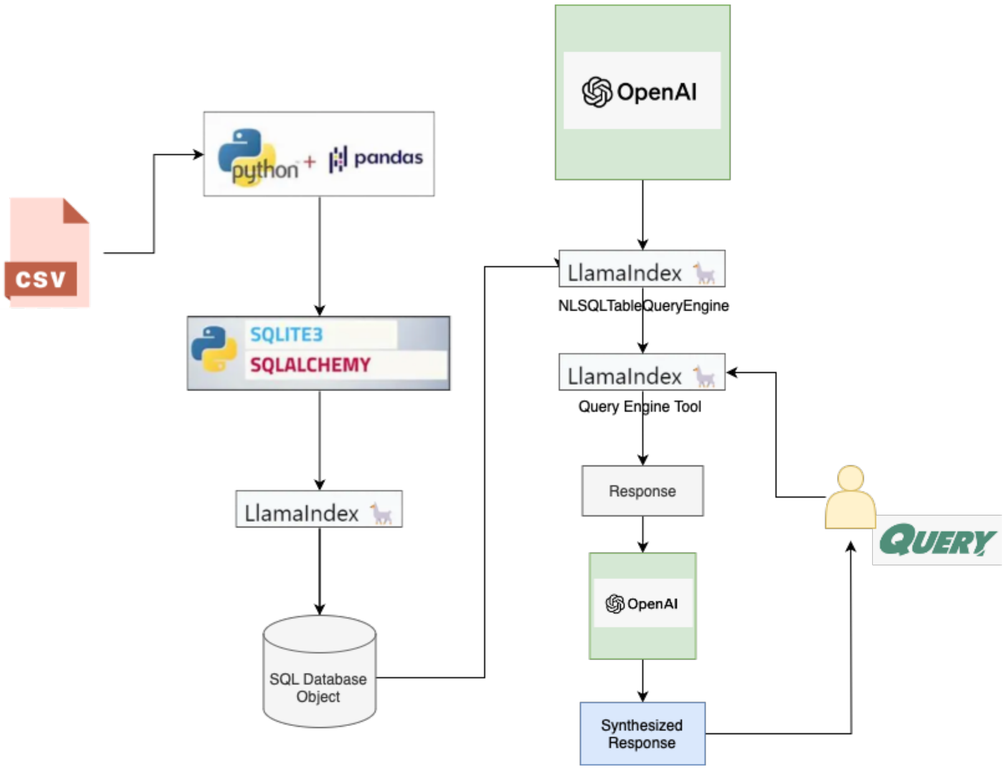

Figure 3.1: Weather Report Generation Pipeline

3.1.1 Data Ingestion

Integrating external data sources with Language Model Models (LLMs) plays an important role in enhancing their contextual understanding improving the accuracy of their responses. In our study, we connect LLMs with domain-specific data sources, particularly weather information, using a structured approach. We accomplish this by integrating LLMs with a SQL database containing weather data sourced from a trusted provider - KNMI[1].

For our project, we aim to ingest CSV data obtained from five distinct weather stations, each providing location-specific weather information. Each CSV file includes a unique column for the weather station code, which corresponds to its respective location, simplifying the identification and reference of location-specific data (e.g., Amsterdam - 240, Rotterdam - 344, etc.).

To integrate CSV files into our LLM application, we begin by reading the data into Pandas DataFrame objects. This step involves loading the CSV file and extracting the relevant data fields. Each column is treated as

[1]Historic data is unfortunately not maintained by Buienradar, but KNMI provides a great alternative resource.

a feature, while each row represents a data entry. By utilizing Pandas, we preprocess the data efficiently, ensuring its quality and consistency.

Once the data has been preprocessed, we create an in-memory SQLite-powered Sqlalchemy engine. This engine serves as the foundation of our data storage and querying system. Each Pandas DataFrame is then converted into a corresponding SQL table within the SQLite database. This conversion process allows us later on in the pipeline to seamlessly integrate and access location-specific weather data within our LLM application.

3.1.2 Indexing

Indexing is the next step in our methodology, enabling efficient querying and retrieval of the data stored in our SQL database. This process allows data accessibility and streamlines the retrieval of information, making it more intuitive for users. It involves transitioning from a conventional SQL database setup to a Table Schema Index, so that during querying, right schema can be retrieved [13].

Each row of the table is processed to generate embeddings. Embeddings are high-dimensional vectors that capture the semantic meaning of the data in each row. These embeddings enable efficient similarity searches. The generated embeddings, along with associated metadata such as the original row index, column names, and any other relevant information, are stored in an index structure called the VectorStoreIndex.

Integration of the VectorStoreIndex with our SQL database is facilitated through the LlamaIndex framework. This integration ensures that the embeddings are stored in a manner that allows our LLMs to access and utilize the semantic content of the data. The SQL Database object encapsulates our SQLite database, seamlessly interfacing it with the LlamaIndex framework. This object not only includes the structured table schemas but also integrates the embeddings, thereby enabling efficient natural language queries.

This indexing process is essential for preparing our data for efficient querying and analysis, laying the groundwork for the subsequent steps in our research. By creating a robust indexing system, we ensure that our LLM application can access and utilize the weather data effectively, enabling us to generate weather reports based on correct data.

3.1.3 Storing Indexes in a Vector Database

A variety of options are available for consideration when it comes to vector databases [23]. For our experiment setup, we selected the Pinecone vector database. It provides a robust and scalable platform for indexing and searching high-dimensional vector data efficiently, necessary for the processing and analyzing our weather data. The choice of Pinecone was driven by its advanced features offering seamless integration with Llamaindex. Its ability to handle large volumes of vector data, even in real-time, coupled with its support for nearest neighbor search and similarity matching, made it a favourable choice. This aligns well with our requirement to perform similarity search on the vector data based on the users' query, and to further analyze large amount of data to provide quick responses for a better user experience.

Configuring our system with Pinecone was straightforward, thanks to its cloud-native architecture that relieved us of any hosting burdens. This allowed us to swiftly store and index vector representations of weather parameters from our dataset, facilitating rapid data retrieval and manipulation during report generation. Setting up indexes and query engines is a key step in utilizing the full functionality of LlamaIndex, enabling to retrieve relevant data.

Pinecone's comprehensive documentation and developer resources helped in simplifying the setup process, allowing us to navigate and leverage its functionalities to meet our project requirements.

3.1.4 Query Engine

With the data indexed and ready for retrieval, query engines comes in facilitating interaction with the data. These engines provide intuitive interfaces for natural language querying, supporting a wide range of applications such as question-answering systems and complex conversational interactions.

When retrieving data, the initial step involves understanding the contextual meaning of user queries, which are expressed in natural language. Subsequently, an SQL query needs to be generated based on this understanding to extract relevant data from the database. The NLSQLTableQueryEngine provides an instantiation for this purpose. It converts natural language queries into SQL queries and executes them against the database using its query() method. The output is a structured list of rows that match the query criteria. To have more control over SQL query generation and response synthesis, we refine the default prompt template for NLSQLTableQueryEngine to better match our application domain. This acts as a set of instructions for the engine to generate queries useful for retrieving the data required our task of report generation of correct location and date. The next section details the development of prompt which not only enhances data accessibility but also streamlines the data analysis process.

3.1.5 Prompt Curation

Prompt engineering can improve the capacity of LLMs [14]. To develop the prompt, it was important to grasp the nuances of interpreting various weather parameters from our dataset. Working closely with the meteorologists to include the instructions guiding the generation of weather reports, we formulated a detailed outline including various factors essential for crafting informative and engaging forecasts. Considerable thought was given to tailoring the prompt to meet the informational needs of users while maintaining a professional yet approachable tone akin to seasoned meteorologists. Striving for a balance between thoroughness and brevity, we delineated specific requirements for content length and structure.

Central to the prompt's design was the inclusion of instructions for accessing the appropriate weather data table based on the users' specified date and location. This ensured the retrieval of relevant and localized information for the given and subsequent day, as we consider one day head for future forecast.

To emulate the writing style of Buienradar, the prompt directed the summarization of daily weather conditions for distinct time periods: morning, afternoon, evening, and night, followed by insights for the following day. Table 3.1 provides the hourly division of each section of the report. Clear instructions were provided for selecting the correct data for each division to ensure accuracy and relevance in the generated reports.

Table 3.1: Hourly division for each section of the report

Report section	Hour division
Today Morning	06:00 - 11:00
Today Afternoon	12:00 - 17:00
Today Evening	18:00 - 23:00
Tonight	00:00 - 05:00
Tomorrow	06:00 - 18:00

Special emphasis was placed on incorporating descriptive language to vividly depict atmospheric phenomena, such as weather, precipitation, temperature, and wind conditions. Additionally, rule-based instructions were integrated to map retrieved parameters to 14 distinct weather conditions, known as weather icons A.1, aligning with the current weather reporting standards. For example, if the data show some sunshine, snow, and there are various precipitation values for the day, the weather conditions will be described as "sunny and snow showers".

Regarding wind description, careful attention was paid to parameters such as wind force and Beaufort scale, drawing insights from authoritative sources to ensure correct wind descriptions. Table 3.2 describes the wind force scale used for this study.

Table 3.2: The Beaufort wind force scale [1]

Wind Force	Description	Wind Speed (Km/h)
0	Calm	< 1
1	Light air	6-11
2	Light breeze	12-19
3	Gentle breeze	12-19
4	Fresh breeze	20-28
5	Moderate breeze	29-38
6	Strong breeze	39-49
7	Near gale	50-61
8	Gale	62-74
9	Strong gale	75-88
10	Storm	89-102
11	Violent storm	103-117
12	Hurricane	118

The prompt encouraged creativity and variety in report writing to sustain user interest and engagement.

This initial prompt was structured to have the query engine retrieve all the data from the requested date and the next date. The engine would then analyze this data from various times of the day and synthesize different sections of the report accordingly, covering the morning, afternoon, evening, night, and the next day. The prompt instructed the inclusion of detailed weather data such as temperature, wind speed and direction, cloud cover, precipitation, and significant weather events for each section of the day.

However, during an internal review, when we compared the temperature values mentioned in the generated report to the retrieved data, we observed that the values were picked randomly from the retrieved data instead of being accurately matched to specific sections of the day. The model struggled to correctly identify and apply the appropriate values for specific hours within each part of the day. This issue was particularly evident when differentiating between the data for morning, afternoon, evening, night, and the next day.

To address this, we refined our approach with a slightly different prompt available in **Appendix A.1**. This revised prompt directed the engine to sequentially analyze data for different times of the day. The process involved the following steps:

1. **Segmented Data Retrieval and Analysis:** The engine was tasked with generating individual reports for each time period—morning, af-

ternoon, evening, night, and the next day. The data was retrieved and analyzed sequentially for each period. First, the engine retrieved data for the morning, analyzed it, and wrote the report for that segment. This process was then repeated for the afternoon, evening, night, and the next day. Each report included specific details such as temperature, wind speed and direction, cloud cover, precipitation, and significant weather events relevant to that particular time segment.

2. **Sequential Data Processing:** By analyzing data in a segmented manner, the engine could more accurately associate the retrieved data with the corresponding time periods. This step ensured that the weather conditions for each part of the day were correctly represented.

3. **Report Synthesis:** After creating the separate reports for each time segment, we used another prompt, as detailed in **Appendix A.2**, to stitch these sections together. This secondary prompt was designed to compile the individual time-specific reports into a cohesive and detailed report for the entire day and extended forecast.

4. **Final Review and Adjustment:** The combined report was then reviewed by meteorologists to ensure consistency and accuracy. Small adjustments were made to maintain the writing style of current weather reports, such as starting each section with "This morning," "This afternoon," "This evening," and so on, as well as rounding up temperature values to make the report sound less robotic.

This multi-step process, known as meta prompting, ensured that the query engine could handle data more effectively, leading to more correct and reliable weather reports. Meta prompting is an advanced technique that allows for a structured approach to interacting with the LLM, emphasizing the structure and syntax of information [9]. By breaking down the data analysis and report generation into smaller, manageable segments, the engine was able to provide a detailed and correct weather forecast for each part of the day. This structured approach improved the overall quality of the generated reports.

3.2 Weather Assistant Design

To extend this project, we build an effective weather assistant to interact with historical weather reports, which uses both structured and unstructured data. We implemented a dual-query system that combines the power of SQL querying with semantic search capabilities. Our approach ensures responses by integrating historical weather reports with rich contextual information. Both the data sources are described as follows:

- Structured Database: Contains historical weather reports that is structured and generated by meteorologists. This data is highly reliable and is used for queries requiring precise historical weather information.

- Unstructured Database: Contains unstructured data, such as articles from Wikipedia. This data provides a broad understanding of climate of the Netherlands and is used for semantic and contextual queries.

The pipeline used for building the weather assistant is visualized in Figure 3.2. The process begins with a user query, which can range from specific historical weather inquiries to broader questions about the climate.

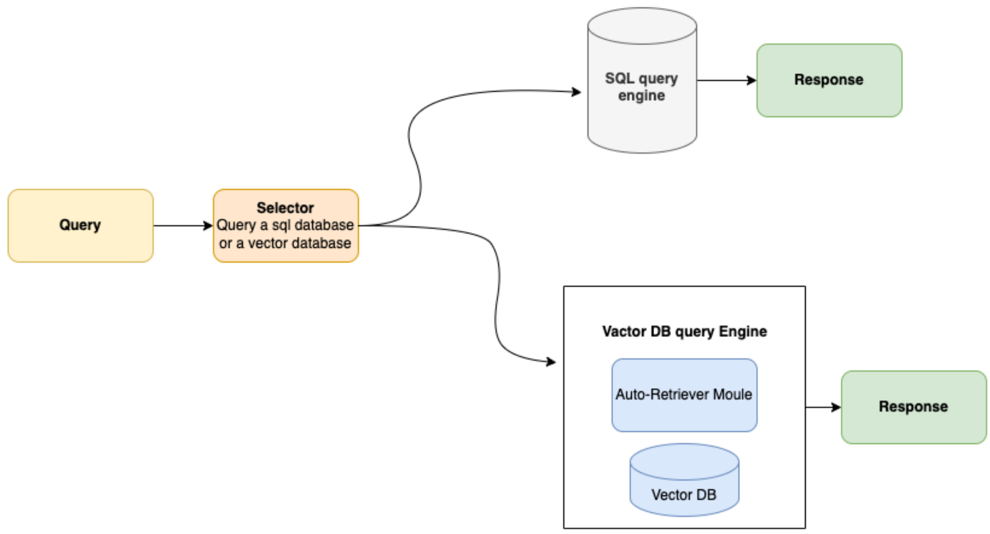

Figure 3.2: Weather Assistant: Pipeline

The query is first processed by the Selector module [15]. This module determines the most appropriate database to query based on the nature of the user query. The Selector uses the LLM to decide whether to route the query to the SQL database or the Vector database, based on the descriptions for the query engines which are as follows:

> SQL Query Engine: "Use this tool to fetch relevant weather reports and analyze them to answer the users' questions effectively."
>
> Vector DB Query Engine: "Ideal for users seeking an understanding of Dutch weather without the need for date-specific weather data."

If the query is directed to the SQL database, the SQL Query Engine is engaged. This engine translates the user's natural language query into an SQL query. The SQL database contains structured data, specifically historical weather reports generated by meteorologists. For our case, we included all the reports only from the year 2023. The engine retrieves relevant weather report from this database, and analyses it to answer the user's query. If the query is better suited for semantic search, the Selector module routes it to the Vector DB Query Engine. This component consists of two main parts:

- Auto-Retriever Module: This module uses similarity search algorithms, like cosine similarity, to compare query vectors with stored vectors in the Vector Database. It identifies the most relevant documents and extracts information to answer the user's query.

- Vector Database: This database holds unstructured data, such as Wikipedia pages and other textual descriptions about the climate of the Netherlands. It stores vector embeddings, which are dense numerical representations of data items like text, images, or other complex types. Website data, such as articles or web pages, is divided into manageable chunks, with each chunk converted into a high-dimensional vector embedding. These embeddings, along with metadata such as document ID and source URL, are stored in the Vector Database, enabling quick and efficient retrieval based on semantic content.

Finally, the system compiles the results from either the SQL Query Engine, the Vector DB Query Engine, or both, and generates a coherent answer. The answer is then presented to the user in a clear and concise manner, to fulfil the query requirements.

The strength of our weather assistant lies in its ability to seamlessly integrate structured and unstructured data. Historical weather data provides precise and factual reports, while descriptive texts offer broader context and explanations. This integration allows users to receive detailed and comprehensive answers to a wide range of weather-related questions.

Chapter 4

Experimental setup

4.1 Dataset

The dataset utilized in this study comprises publicly available historical weather data obtained from the Royal Dutch Meteorological Institute (KNMI)[11], a governmental agency responsible for reporting current weather conditions. All data is collected at the KNMI data center, which is part of the World Meteorological Organisation's global network of datacenters. It maintains 50 weather stations across the Netherlands, each equipped with instruments for measuring various meteorological parameters.

We retrieved the data for the year 2023 from the KNMI website's hourly data repository[10] in CSV format. For the purpose of this study, five distinct weather stations were selected for analysis, namely De Bilt, Schiphol, Eindhoven, Rotterdam, and Maastricht. These stations represent diverse geographical locations within the Netherlands, offering variations across different regions.

The dataset has a wide range of meteorological variables, including but not limited to temperature, precipitation, wind speed, and cloud cover. Each observation in the dataset corresponds to a specific hour, providing a detailed temporal resolution for the analysis. A sample of the raw CSV data from one of the weather stations can be found in **Appendix A.5**.

4.2 Data Pre-processing

Currently, Buienradar utilizes the forecast data using an API call, based on which the report is written. Due to the unavailability of historical forecast data from Buienradar, we turn to KNMI for our historical weather data needs. This section outlines the steps taken to preprocess the weather data, discussing the transformations applied to enhance the quality and usability of the dataset.

To standardize the temperature values, we perform a conversion from 0.1 degrees Celsius to 1 degrees Celsius. This ensures uniformity in temperature reporting across our dataset. Wind speeds are transformed into the Beaufort Scale, an empirical measure that relates wind speed to observed conditions at sea or on land. We refer to authoritative sources [1] for the conversion process, enhancing the interpretability of wind speed data. Additionally, wind directions are initially recorded in degrees. To facilitate better comprehension, we convert these angular measurements into cardinal directions.

To contextualize our analyses, we incorporate data on the average temperature in the Netherlands for each month. This additional information can be useful in the identification of extreme temperature events. By undertaking these preprocessing steps, we ensure the consistency, comparability, and interpretability of the weather data.

4.3 Experiment Details

All experiments for the weather report generation system and the weather assistant were conducted on the Databricks platform, leveraging Azure Storage for secure and efficient data management. This setup provided a robust and scalable environment to execute our system effectively. The implementation of the methods described in this thesis can be found in the code repository WeatherReportingSystem [1].

- Computational Resources:

  Single driver with 16 GB of memory and 4 cores.

  Runtime 14.3 LTS for Machine Learning with Scala 2.12 [6]. This environment includes essential libraries and tools, and provides the necessary computational resources facilitating the implementation and testing of Llamaindex framework for generating weather reports. This setup ensures efficient resource management and scalability for deployment.

- Language Model:

  Azure OpenAI's GPT-3.5-turbo [16], integrated to leverage its advanced text generation capabilities within the LlamaIndex framework.

- Embeddings:

[1] https://github.com/sonimegha123/WeatherReportingSystem.git

Azure OpenAI's text-embedding-ada-002 model [16], used to generate high-quality embeddings, essential for capturing semantic information in the LlamaIndex-based system.

- Vector Database:

  Pinecone [2], utilized as a vector database to store the embeddings, making them efficiently accessible and understandable by the LLM. Pinecone has following key benefits [25]:

  - Ultra-low query latency at any scale, ensuring fast and efficient retrieval of embeddings.
  - Combines state-of-the-art vector search methods enhancing the relevance and accuracy of search results.
  - Live index updates, allowing for real-time addition, editing, or deletion of data without downtime.
  - Scalability to handle very large datasets, including hundreds of millions to billions of vector embeddings.
  - Fully managed service, providing ease of use and scalability.

By utilizing these resources, we ensured the system's performance and scalability, facilitating efficient processing and generation of weather reports.

In order to answer **research question 1**—What do users in the Netherlands expect from Buienradar regarding daily weather insights for specific locations?—we mainly rely on insights from the Buienradar team, who create the weather reports for their website.

According to their observations, the majority of users, approximately 70%, visit the Buienradar website to check for rain predictions for the next three hours. This allows them to plan their activities accordingly, such as deciding whether to carry a jacket or umbrella to avoid getting caught in the rain. Additionally, about 20% of the users visit the site to check the 14-day forecast. This longer-term outlook helps them plan their week, enabling decisions like scheduling a camping trip or planning other outdoor activities. The remaining 10% of users access specific maps available on the platform, such as those showing pollen density or wind speeds. While these numbers are approximations based on trends observed by the Buienradar team over the years, they provide valuable insights into the varying needs and expectations of users seeking weather information for specific locations within the Netherlands.

[2] https://www.pinecone.io

These insights highlight the importance of providing detailed and timely local weather reports that cater to the immediate and future planning needs of users. For our project, this means structuring our weather reports into distinct sections that allow users to directly jump to the information they need, without having to read through the entire report. Each section delivers precise and relevant forecasts for different parts of the day, addressing the immediate need for short-term weather predictions.

Additionally, we include a comprehensive section at the end for future predictions, focusing on one day in advance. This section is crucial for users planning their activities for the next day. By aligning our report structure with user expectations, we aim to enhance the usefulness and accuracy of our local weather reports. This ensures they effectively support daily decision-making processes for users across different geographical locations within the Netherlands, allowing for quick and easy access to the most pertinent weather information.

To address **research question 2**, which focuses on understanding how individuals perceive weather reports generated by Artificial Intelligence (AI) systems, we set up the following experiment to elicit user feedback and perceptions. Firstly, we develop an AI-driven weather forecasting system capable of generating forecasts based on historical weather data. This system serves as the basis for our experiment, allowing us to provide participants with AI-generated weather reports for evaluation. Participants in the study included 9 individuals who work in the annotation lab to label data. They are all native Dutch speakers. These participants were presented with a number of weather forecasts generated by our AI-driven system, accompanied by the underlying data retrieved by our system as the basis for the generated reports. Prior to evaluating the forecasts, participants were instructed to review a sample of current weather reports from Buienradar to familiarize themselves with the existing style and content of hand-written reports on the platform. Subsequently, participants were presented with a set of questions aimed at evaluating various aspects of the AI-generated reports. They were asked to assess the accuracy of the forecasts by comparing them to the retrieved data, as well as to investigate their adherence to the Buienradar-style weather reports. Questions included inquiries about the ease of understanding of the report, accuracy, the likelihood of identifying it as automatically generated, and overall satisfaction with the report's quality.

In designing the survey, we ensured it comprehensively covered key aspects of the AI-generated weather reports. The survey included questions aimed at evaluating specific elements such as the parts of the day covered by the forecast, the types of weather information provided (e.g., temperature, precipitation, cloud cover), the accuracy of the displayed data, and the likelihood of identifying it as automatically generated. Participants were also asked to rate the overall quality of the weather reports on a scale and to assess the ease of understanding and the design layout of the forecasts. The design process took into account the need for both quantitative and qualitative feedback. To ensure the survey's effectiveness, we conducted a pilot study with a small group of participants from within the team. This pilot phase allowed us to test the clarity of the questions, identify any potential ambiguities, and ensure that the survey was easy to follow. Feedback from the pilot survey was used to make necessary adjustments, enhancing the validity of the final questionnaire. The actual questions can be found in **Appendix A.3**.

These reports and the accompanying questionnaire were presented to the AnnotationLab team, using the Labelbox tool. It allows full visibility into evaluation activity and progress. By examining user attitudes, beliefs, and opinions towards AI-generated reports, we aim to shed light on the level of acceptance and confidence users place in such systems. Additionally, we consider inter-annotator agreement when evaluating the AI-generated weather reports, as it reflects the consistency with which different annotators assess the same reports.

To address **research question 3**, we aim to explore the feasibility of using the same approach employed for generating weather reports to create a personalized weather chat experience. We only informally evaluate a number of representative test questions that users may issue to the chatbot.

4.4 Experimental Results

4.4.1 Generated weather reports

The study aims to assess how users perceive AI-generated weather reports in terms of their authenticity and human-like qualities, satisfaction levels, report structure, and language clarity. Before delving into the evaluation, it is worth noting that example reports, generated in both English and Dutch, can be found in **Appendix A.4**. An analysis of annotator responses revealed following insights:

- **Auto-Generation Recognition:**

  The distribution of ratings indicates that most annotators can detect auto-generated content to varying degrees. The peak at rating 3 suggests that the most common perception is that the

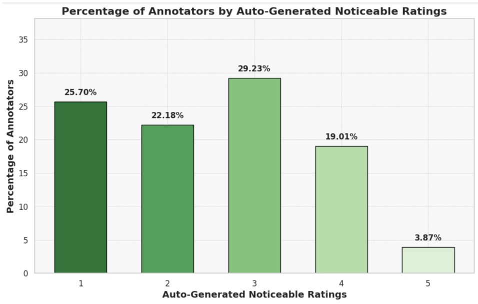

Figure 4.1: Distribution of Annotator Auto-Generation Recognition Ratings (1 = Auto-generated, 5 = human-written), showing the percentage of annotators for each recognition level.

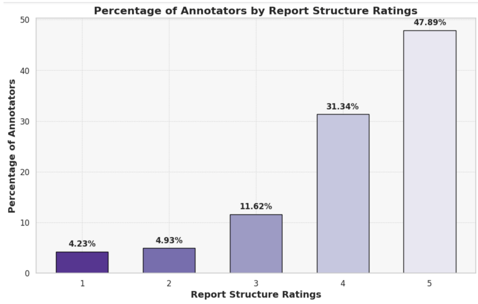

Figure 4.2: Distribution of Annotator Report Structure Ratings (1 = lowest, 5 = highest), showing the percentage of annotators for each clarity level.

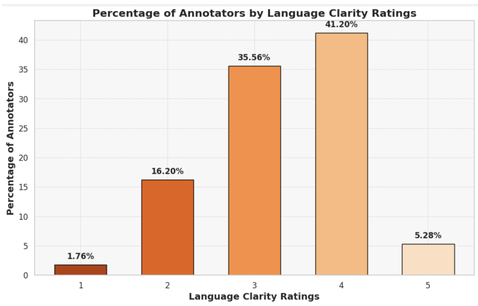

Figure 4.3: Distribution of Annotator Language Clarity Ratings (1 = lowest, 5 = highest), showing the percentage of annotators for each clarity level.

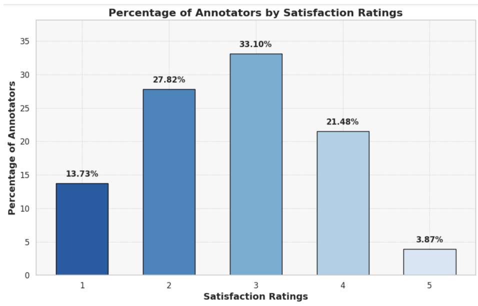

Figure 4.4: Distribution of Annotator Satisfaction Ratings (1 = lowest, 5 = highest), showing the percentage of annotators for each satisfaction level.

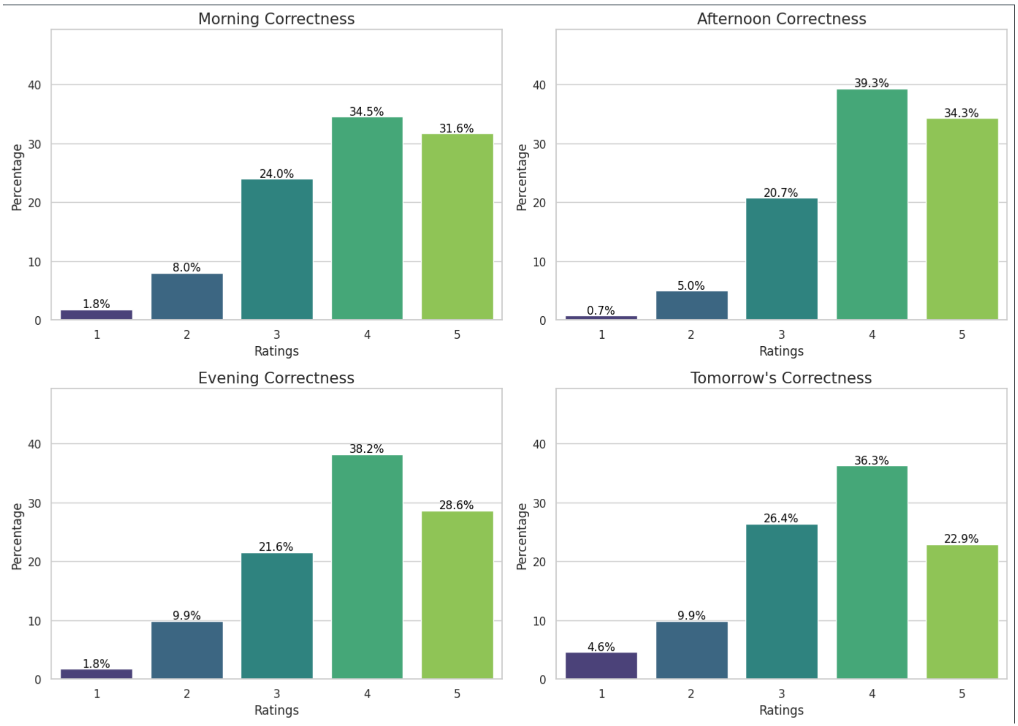

Figure 4.5: Distribution of Annotator Satisfaction Ratings (1 = lowest, 5 = highest), showing the percentage of annotators for Correctness in AI-Generated Content.

content falls in a grey area where it is not easily distinguishable as either completely auto-generated or human-written (see Figure 4.1). In contrast, only 23% believed the reports were “Mostly or surely Human Written”. This disparity highlights the prevalent recognition of AI's involvement in report generation.

Interpretation: These results indicate a need for enhanced natural language generation capabilities to improve authenticity and human-like qualities in the reports.

- **Report Structure Rating:**

  Most annotators (79.23%) rated the structure of AI-generated weather reports as either “Good” or “Very Good” (see Figure 4.2). Only 9.16% rated them as “Bad” or “Very Bad”. Annotators noted that almost all reports included comprehensive information for morning, afternoon, evening, and the next day.

  Interpretation: This high level of positive feedback on structure indicates generally favorable perceptions of report organization and layout. Our system successfully adhered to the demanded report structure, maintaining the current reporting style.

- **Language Clarity:**

  Feedback on language clarity was mixed (see Figure 4.3):

  * 46.48% rated the language as "Well Understood" or "Perfectly Understandable".
  * 17.96% rated it as "Unclear".

  Interpretation: While nearly half found the language clear, a minority noted room for improvement in clarity.

- **Report Satisfaction:**

  Evaluators satisfaction ratings varied considerably(see Figure 4.4):

  * 60.92% rating satisfaction levels 3 and 2.
  * 21.48% rating satisfaction level 4, indicating a satisfactory but not exceptional quality.
  * 13% expressing very unsatisfactory experiences at level 1.
  * Only 3.87% rating satisfaction at the highest level of 5.

  Interpretation: Most evaluators (60.92%) rated the reports as moderate (level 3) or below (level 2), suggesting room for improving report quality to achieve higher satisfaction. The low percentage (3.87%) of exceptional ratings suggests need to elevate overall satisfaction.

- **Report Correctness:**

  The evaluation analyzed annotator feedback on report correctness across different times of the day: morning, afternoon, evening, and predictions for the next day. Annotators rated the correctness on a scale of 1 to 5, where 1 indicates the lowest and 5 indicates the highest.

  Interpretation: The distribution of these ratings is shown in Figure 4.5. Most ratings fall within the higher end of the scale, indicating a generally positive reception and suggesting that the AI-generated weather reports are considered trustworthy.

The evaluation of AI-generated weather reports reveals a nuanced perception among users. While there is recognition of AI involvement and generally positive feedback on structure, satisfaction levels vary, particularly regarding language clarity. These findings underscore the importance of refining AI capabilities to enhance authenticity and user satisfaction in weather forecasting. The data indicates a high level of satisfaction with the AI-generated weather reports across all times of the day, with the vast majority of ratings being 3 or higher. This suggests that the AI system provides reliable weather information, earning trust from the annotators.

When evaluating AI-generated weather reports, it's crucial to consider inter-annotator agreement, which reflects how consistently different annotators assess the same reports. High consensus among annotators indicates reliable and clear evaluation criteria, enhancing the credibility of the evaluation results. Conversely, low agreement suggests varied interpretations and the need for better-defined evaluation standards.

The Labelbox tool was used for the evaluations but has limitations in measuring consensus, particularly when more than two annotators are involved. Labelbox's agreement calculations for radio classifications and checklist classifications are straightforward but insufficient for multiple raters. To address the limitations of Labelbox, we used Krippendorff's alpha [21]. This metric generalizes interrater reliability to any number of annotators and various data types. Krippendorff's alpha ranges from -1 to 1:

1: Perfect agreement

0: Agreement is no better than chance

-1: Systematic disagreement among annotators

Table 4.1: Consensus scores for different aspects of the AI-generated weather reports. -1(systematic disagreement), 0(random), 1(perfect agreement)

Aspects	Consensus Scores
AI-Generated Noticeable	-0.0617
Report structure	-0.0353
Language clarity	-0.0334
Overall satisfaction	-0.0494
Morning weather accuracy	-0.0359
Afternoon weather accuracy	-0.0537
Evening weather accuracy	-0.0270
Night weather accuracy	-0.0331
Tomorrow's weather accuracy	-0.0017

Table 4.1 shows the Krippendorff's alpha values for different aspects of the AI-generated weather reports. The negative values suggest that annotators systematically disagreed in their evaluations.

- **AutoGeneratedNoticeable:** Annotators have different criteria for recognizing auto-generation, indicating varied perceptions of what makes a report appear auto-generated.
- **ReportStructureGood:** There is no consensus on the quality of the report structure, reflecting differing opinions on what constitutes a well-structured report.
- **LanguageClarity:** The lack of agreement on language clarity suggests that annotators have diverse standards for what they consider clear and understandable language.
- **Satisfaction:** Annotators do not agree on their overall satisfaction with the reports, highlighting a range of expectations and experiences with the AI-generated content.
- **Accuracy:** The disagreement on the correctness of the weather reports across different times of the day suggests varied standards or interpretations of what constitutes accurate weather reporting.

Although these values are negative, they are closer to 0, indicating that while there is systematic disagreement, it is not highly polarized. This suggests that the differences in perception among annotators are not extreme but rather reflect individual preferences and priorities. It might be that people perceive the weather reports differently based on what matters most to them, such as clarity, structure, or perceived accuracy.

4.4.2 Weather Assistant Performance

In order to see whether the RAG approach used for generating weather reports can also be adapted to develop a personalized weather chat experience, we conducted an informal evaluation using questions that users might ask the chatbot. Below are the responses generated by our weather assistant to some sample queries:

➔ **User:** What was the weather like on August 15, 2023?
Bot: On August 15, 2023, vacationers in the Netherlands enjoyed sunny weather with temperatures reaching summer levels. There were some clouds in the morning, but it cleared up in the afternoon. There was a chance of mist forming overnight, especially in the northern half of the country. Overall, it was a good day for outdoor activities.

➔ **User:** Should I wear a jacket on June 6, 2023?
Bot: Based on the weather report for June 6, 2023, it seems that there will be warm and possibly even tropical temperatures. It is unlikely that you will need to wear a jacket on that day.

➔ **User:** What is the typical weather in the Netherlands in November?
Bot: The weather in the Netherlands during Autumn can vary, but November is typically a chilly month with temperatures ranging from 3 to 9 degrees Celsius. It is also known to be a rainy month, so it is recommended to bring warm and waterproof clothing when visiting during this time.

➔ **User:** Is it a good day to for a barbecue on July 10, 2023?
Bot: Based on the weather data provided for July 10, 2023, it appears to be a good day for a barbecue as the sun will be out and it will be less humid and warm than the previous weekend. However, there may be some scattered clouds later in the day.

➔ **User:** Was it windy on October 31, 2023?
Bot: Yes, it was windy on October 31, 2023. The weather report for that day mentions a lot of wind and a storm depression.

➔ **User:** Can you plan a day for activities to do in December?
Bot: Yes, there are many activities that can be planned for December in the Netherlands during winter, including ice skating, visiting Christmas markets, exploring museums, and enjoying warm drinks at cozy cafes. However, it may be best to plan indoor activities for most of December due to the changeable weather with regular rain or showers. It is important to keep in mind that the weather can change quickly, so it may be best to have a backup plan in case of inclement weather. There may also be some opportunities for outdoor activities such as ice skating or holiday light displays on days when the weather is expected to be drier.

➔ **User:** What is usually the minimum temperature in the Netherlands during winter?
Bot: There is no available data or information provided to answer your question.

These queries demonstrate the range of information our weather assistant can provide. It utilizes both the data sources, historical weather reports and Wikipedia, to answer users' questions and refrains from generating random responses when lacking context.

Chapter 5

Conclusion

5.1 Summary

The evaluation of AI-generated weather reports provided significant insights into user perceptions and the overall effectiveness of the AI system. The study aimed to assess how users perceive AI-generated weather reports in terms of their authenticity, structure, language clarity, and satisfaction. The results highlight the complex interplay of user expectations, biases, and the inherent challenges of creating human-like AI content.

One key finding is the systematic disagreement among annotators, as indicated by the consensus scores across various dimensions such as report authenticity, structure, language clarity, and satisfaction. Although the Krippendorff's alpha values are negative, their proximity to zero suggests that while annotators did not agree, their disagreements were not highly polarized. This likely reflects the diverse priorities and individual preferences of the annotators when evaluating the reports.

It is also important to consider the context in which these evaluations were conducted. Annotators were aware that the reports were AI-generated and before evaluating, they were asked to read the example report on Buienradar website to get an idea of what a human-written report can look like. This prior knowledge might have introduced a bias, making annotators more inclined to perceive the AI reports as less authentic and more machine-like. This bias could have influenced their evaluations, leading to a higher recognition of AI involvement and possibly lower ratings for aspects such as language clarity and satisfaction.

Additionally, because the annotators were comfortable with Dutch language, the reports were generated in Dutch. It might have been the case that reports in English would have received better language clarity ratings, as the translation processes and language nuances can affect the clarity and perceived quality of the content.

Overall, the findings suggest that while AI-generated weather reports are generally well-received, there is room for improvement, particularly in making the reports more human-like and satisfying to a broader audience.

Applying the same AI-driven approach used for generating reports to create an interactive weather chat experience seems to be effective. The assistant demonstrated its ability to interact with historical weather reports, addressing both specific historical weather inquiries and broader climatic questions. This approach showcase the capability to provide comprehensive and concise responses, offering a personalized and informative user experience.

5.2 Discussion

In evaluating AI-generated weather reports, it's important to define what constitutes a 'good enough' weather report. This concept varies among users, but generally, a 'good enough' report should provide accurate, clear, and relevant information that users can easily understand and act upon. The lack of a standardized benchmark poses a challenge in assessing the effectiveness of AI-generated weather reports, particularly since the use of LLMs in weather and climate forecasting is still relatively new. Without a standardized benchmark, assessing the effectiveness and quality of AI-generated content remains subjective. Furthermore, the current human-written reports, found on Buienradar platform, provide a broad climatic overview of the entire country, serving different purposes than AI-generated reports that focus on specific, localized forecasts. This difference in scope and intent makes direct comparisons between human and AI-generated reports less meaningful. Therefore, while our study provides valuable insights into the user perception of AI-generated weather content, understanding these nuances is crucial for accurately evaluating and improving AI-generated weather reports to better meet user needs and expectations.

5.3 Future Work

While our current study focuses exclusively on weather data from the year 2023, there are several promising directions for future research that could enhance the comprehensiveness and real-time applicability of our weather report generation system.

Firstly, it would be interesting to explore how incorporating real-time weather predictions could improve the relevance and immediacy of the generated reports. By integrating an API call to fetch the current day's weather data, our system might produce up-to-the-minute weather forecasts, providing users with real-time insights. This enhancement could ensure that users receive the most current and accurate weather information possible, aligning closely with their daily needs and expectations.

Additionally, expanding our dataset to include historical weather data spanning multiple years could offer valuable context on weather patterns and changes over time. Analyzing long-term trends and anomalies in historical data would not only enrich the generated reports but also enable more informed predictions and comparisons. It would be nice to see if this broader dataset can provide a more robust foundation for understanding seasonal variations and long-term climate changes, ultimately enhancing the predictive power of our system. For example: "The last occurrence of such warm temperatures this early in the year was in [last year — 1965 — 1887]".

Furthermore, it would be beneficial to see how the functionality of the weather assistant could be enhanced to operate on real-time local weather reports. Currently, the assistant provides responses based on historical reports. Transitioning to real-time data might make the weather assistant more dynamic and responsive, thereby increasing its utility for users seeking immediate and localized weather information. This upgrade could transform the assistant into a more effective tool for real-time decision-making and planning

Exploring these future possibilities could lead to a more responsive, accurate, and comprehensive weather reporting system that better serves the needs of users across different geographical locations.

Bibliography

[1] Beaufort Wind Scale. https://www.rmets.org/metmatters/beaufort-wind-scale. Accessed: 2024.

[2] Buienradar-RTL Netherlands. https://www.buienradar.nl. Accessed: 2024.

[3] Zhen Bi, Ningyu Zhang, Yida Xue, Yixin Ou, Daxiong Ji, Guozhou Zheng, and Huajun Chen. Oceangpt: A large language model for ocean science tasks. *arXiv preprint arXiv:2310.02031*, 2023.

[4] Shengchao Chen, Guodong Long, Jing Jiang, Dikai Liu, and Chengqi Zhang. Foundation models for weather and climate data understanding: A comprehensive survey. *arXiv preprint arXiv:2312.03014*, 2023.

[5] Grace Colverd, Paul Darm, Leonard Silverberg, and Noah Kasmanoff. FloodBrain: flood disaster reporting by web-based retrieval augmented generation with an LLM. In *Artificial Intelligence for Humanitarian Assistance and Disaster Response Workshop*, 2023.

[6] Databricks. Databricks Runtime. https://learn.microsoft.com/en-us/azure/databricks/release-notes/runtime/14.3lts-ml, Accessed: 2024.

[7] Yujuan Ding, Wenqi Fan, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. A Survey on RAG Meets LLMs: Towards Retrieval-Augmented Large Language Models. *arXiv preprint arXiv:2405.06211*, 2024.

[8] Anders Doksæter Sivle and Stein Dankert Kolstø. Use of online weather information in everyday decision-making by laypeople and implications for communication of weather information. *Meteorological Applications*, 23(4):650–662, 2016.

[9] Yutai Hou, Hongyuan Dong, Xinghao Wang, Bohan Li, and Wanxiang Che. MetaPrompting: Learning to Learn Better Prompts. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3251–3262, 2022.

[10] KNMI. Hourly Data Repository. `https://daggegevens.knmi.nl/klimatologie/uurgegevens`. Accessed: 2024.

[11] KNMI. Royal Dutch Meteorological Institute. `https://www.knmi.nl/over-het-knmi/about`. Accessed: 2024.

[12] Jerry Liu. Llamaindex. `https://docs.llamaindex.ai/en/stable/`. Accessed: 2024.

[13] Jerry Liu. llamaindex structured data. `https://docs.llamaindex.ai/en/stable/understanding/putting_it_all_together/structured_data/`. Accessed: 2024.

[14] Ggaliwango Marvin, Nakayiza Hellen, Daudi Jjingo, and Joyce Nakatumba-Nabende. Prompt engineering in large language models. In *International conference on data intelligence and cognitive informatics*, pages 387–402. Springer, 2023.

[15] Horia Modran, Ioana Corina Bogdan, Doru Ursuțiu, Cornel Samoila, and Paul Livius Modran. LLM Intelligent Agent Tutoring in Higher Education Courses using a RAG Approach. *Preprints*, 2024.

[16] OpenAI. AzureOpenAI. `https://learn.microsoft.com/en-us/azure/ai-services/openai`, Accessed: 2024.

[17] Mohaimenul Azam Khan Raiaan, Md Saddam Hossain Mukta, Kaniz Fatema, Nur Mohammad Fahad, Sadman Sakib, Most Marufatul Jannat Mim, Jubaer Ahmad, Mohammed Eunus Ali, and Sami Azam. A review on large Language Models: Architectures, applications, taxonomies, open issues and challenges. *IEEE Access*, 2024.

[18] Alexander Rau, Stephan Rau, Daniela Zoeller, Anna Fink, Hien Tran, Caroline Wilpert, Johanna Nattenmueller, Jakob Neubauer, Fabian Bamberg, Marco Reisert, et al. A context-based chatbot surpasses radiologists and generic ChatGPT in following the ACR appropriateness guidelines. *Radiology*, 308(1):e230970, 2023.

[19] Michelle Rutty and Jean Andrey. Weather forecast use for winter recreation. *Weather, Climate, and Society*, 6(3):293–306, 2014.

[20] David M Schultz, Eve C Gruntfest, Mary H Hayden, Charles C Benight, Sheldon Drobot, and Lindsey R Barnes. Decision making by Austin, Texas, residents in hypothetical tornado scenarios. *Weather, Climate, and Society*, 2(3):249–254, 2010.

[21] Bizhan Shabankhani, Jamshid Yazdani Charati, Keihan Shabankhani, and Saeid Kaviani Cherati. Survey of agreement between raters for nominal data using krippendorff's Alpha. *Archives of Pharmacy Practice*, 11(1-2020):160–164, 2020.

[22] Heydar Soudani, Evangelos Kanoulas, and Faegheh Hasibi. Fine Tuning vs. Retrieval Augmented Generation for Less Popular Knowledge. *arXiv preprint arXiv:2403.01432*, 2024.

[23] The Data Quarry::blog - Prashanth Rao. Vector db, Accessed: 2024.

[24] Nicolas Webersinke, Mathias Kraus, Julia Bingler, and Markus Leippold. CLIMATEBERT: A Pretrained Language Model for Climate-Related Text. *Available at SSRN 4229146*, 2022.

[25] Xie, Xingrui and Liu, Han and Hou, Wenzhe and Huang, Hongbin. A Brief Survey of Vector Databases. In *2023 9th International Conference on Big Data and Information Analytics (BigDIA)*, pages 364–371. IEEE, 2023.

[26] Bruno Zirnstein. Extended context for InstructGPT with LlamaIndex, 2023.

Appendix A

Appendix

A.1 Prompt 1:

Generate a weather report for the given location and given time in the Netherlands. Provide a comprehensive overview of the weather conditions for the specified time. Ensure that the report is written in a professional tone, resembling that of a meteorologist delivering the forecast with warmth and approachability. Keep the information no longer than 3 sentences. Include variations while writing the report to keep it interesting. Do not mention numerical cloud cover values directly in the report.

For the location and given time, analyze the weather data to create a detailed report. Begin by summarizing the overall weather forecast for the given time in a paragraph that paints a vivid picture of what to expect.

Follow the provided example format for the weather report:
Start by introducing the given time, and then give some info about temperature, precipitation and wind. Paint a vivid picture of what to expect. Ensure that the report is written in a friendly tone, resembling that of a meteorologist delivering the forecast with warmth and approachability. Be creative and include variations while writing the report to keep it interesting.

Clouds and Precipitation:
Give an indication of how cloudy the day will be, based on cloud cover values. Instead of numerical values for cloud cover, use descriptive terms derived from the data to illustrate the cloudiness and sunshine.
Consider the following parameters to assess precipitation and provide descriptive weather descriptions in the report accordingly:
- SQ (Duration of sunshine per hourly period): Use SQ values to determine the presence of sunshine and its duration.

- N (Cloud cover): Assess cloud cover based on N values(0=no clouds, 9=sky invisible) and provide descriptive weather descriptions accordingly.
- R (Rainfall): Determine the occurrence of rainfall based on R values.
- RH (Hourly precipitation amount)(Relative Humidity (Hourly precipitation amount (in 0.1 mm) (-1 for < 0.05 mm) / Hourly precipitation amount (in 0.1 mm) (-1 for < 0.05 mm)))
- O (Thunderstorm (0=did not occur; 1=occurred in the previous hour))
- S (Snowfall (0=did not occur; 1=occurred in the previous hour))
- M (Mist (0=did not occur; 1=occurred in the previous hour))

Based on the parameters, describe the sunshine and cloudiness as follows:
Look at values of sunshine, cloudcover, rain, snow, fog, thunderstorm, Relative Humidity values to describe the weather conditions as follows:
Sunny, Gray, Foggy, sun and showers, Thunderstorms, sun and cumulus clouds, occasional rain, fog followed by sun, rain, hardly sunny, cumulus clouds and thunderstorms, snow showers, sun ans snow showers, and winter showers.
- if sunshine values indicate ample sunshine, describe as "Sunny afternoon" or "Sunny day".
- for high cloudcover and low sunshine values describe as "Gray day".
- if fog equals 1 describe as "Foggy".
- for moderate sunshine, high cloudcover, and rain equals 1, describe as "Sun and showers".
- if thunderstorm equals 1, rain equals 1, and sunshine is non 0 with some cloudcover, describe as "Thunderstorms".
- for non 0 sunshine and some cloudcover, describe as "Sun and cumulus clouds".
- if rain equals 1, various precipitation values, and sunshine is low or 0, describe as "Occasional rain".
- if fog equals 1 and sunshine increases with time describe as "Fog followed by sun".
- for rain equals 1, describe as "Rain".
- for high cloudcover and low or no sunshine describe as "Gray and hardly sunny".
- for some sunshine, thunderstorm equals 1, and some cloudcover, describe as "Cumulus clouds and thunderstorms".
- if snow equals 1, and various precipitation values, describe as "Snow showers".
- for some sunshine, snow equals 1, and various precipitation values describe as "Sun and snow showers".
- if low or 0 sunshine, rain equals 1, snow equals 1, and various precipitation values describe as "Winter showers".

Temperature (temperature in celsius):
Discuss the temperature variations throughout the day, analyzing data from the "temperature" column. Pick the right temperature values for both the dates.

Wind:
Describe the wind conditions using the Beaufort wind force scale, taking into account the following features:
- DD (Wind Direction)
- FH (Beaufort wind force)
Discuss the wind force descriptions based on Beaufort scale values (FH):
0: Calm, 1: Light Air, 2: Light Breeze, 3: Gentle Breeze, 4: Moderate Breeze, 5: Fresh Breeze, 6: Strong Breeze, 7: Near Gale, 8: Gale, 9: Strong Gale, 10: Storm, 11: Violent Storm, 12: Hurricane.
Ensure to describe wind conditions in a descriptive and vivid manner, utilizing the Beaufort scale descriptions to convey the strength and impact of the wind.

Table Selection:
Ensure to utilize the correct table named after the location requested by the user to fetch the weather data. If the location is Eindhoven, use the table with station value 370. If the location is De Bilt, use the table with station value 260. If the location is Amsterdam, use the table with station value 240. If the location is Maastricht, use the table with station value 380. And if the location is Rotterdam, use the table with station value 344. The dates are in the format YYYYMMDD in the tables.

Please consider the following hour for each section of the report:
When talking about today, look at the data only for the given date and given location.
- When talking about today morning, look at the rows with Time_category value "Morning" for the given date and given location.
- When talking about today afternoon, look at the rows with Time_category value "Afternoon" for the given date and given location.
- When talking about today Evening, look at the rows with Time_category value "Evening" for the given date and given location.
- When talking about today Night, look at the rows with Time_category value "Night" for the given date and given location.
- When talking about the full day, look at all Time_category values that are not Null for the given day and given location.

SELECT Time_Category, temperature, winddirection, beaufort, precipitation, sunshine, cloudcover, fog, rain, snow, thunderstorm FROM [location] WHERE date = [date] AND Time_Category = [time_category]

or

SELECT Time_Category, temperature, winddirection, beaufort, precipitation, sunshine, cloudcover, fog, rain, snow, thunderstorm FROM [location] WHERE date = [date] OR Time_Category IS NOT NULL

A.2 Prompt 2:

Translate the weather data for given location and day to create a detailed weather report in Dutch language. Do not include dates in the weather forecast.

Note: Focus on highlighting any changes in weather conditions during the day. If conditions remain consistent, avoid repeating details unnecessarily. Do not provide numerical values for cloud cover.

Below you will find the English weather reports:
The morning is described here:
{morning}

The afternoon is described here:
{afternoon}

The evening is described here:
{evening}

The night is described here:
{night}

The next day is described here:
{tomorrow}

Start by summarizing the general weather forecast in two sentences so that the reader gets a vivid idea of what to expect without mentioning a specific time of day. This should be followed by a morning paragraph starting with "In the morning", which brings the weather conditions to life. Then provide an afternoon paragraph starting with "This afternoon" or "In the afternoon", and describe this in the same light-hearted descriptive tone. Then describe the evening and the night together in a paragraph that describes the weather in the evening and night in a fascinating way. Finally, tell a few lines about the next day, starting with "Tomorrow," keeping the tone light and the descriptions lively.

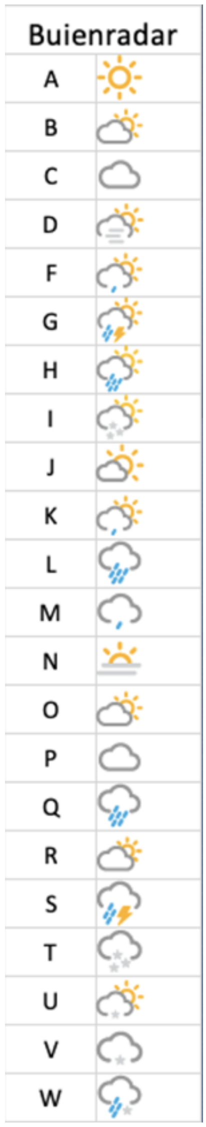

Figure A.1: Weather Icons Depicting Various Weather Conditions. Currently, Buienradar utilizes 14 of these icons for their weather reporting.

A.3 Evaluation Questionnaire:

1.
   - Question: Is information provided about this morning?
   - Options:

1.Yes
2. No

2.
   - Question: What weather information is given for this morning?
   - Options:
     1. Temperature
     2. Precipitation
     3. Cloud Cover
     4. Wind
     5. Thunderstorm
     6. Snow

3.
   - Question: How correct is the information about this morning on a scale of 1 to 5?
   - Options:
     1. Completely inaccurate
     2.
     3.
     4.
     5. Completely Accurate

4.
   - Question: Is information provided about this afternoon?
   - Options:
     1. Yes
     2. No

5.
   - Question: What weather information is given for this afternoon?
   - Options:
     1. Temperature
     2. Precipitation
     3. Cloud Cover
     4. Wind
     5. Thunderstorm
     6. Snow

6.
   - Question: How correct is the information about this afternoon on a scale of 1 to 5?
   - Options:
     1. Completely inaccurate
     2.

   3.
   4.
   5. Completely Accurate

7. - Question: Is information provided about this evening?
   - Options:
     1. Yes
     2. No

8. - Question: What weather information is given for this evening?
   - Options:
     1. Temperature
     2. Precipitation
     3. Cloud Cover
     4. Wind
     5. Thunderstorm
     6. Snow

9. - Question: How correct is the information about this evening on a scale of 1 to 5?
   - Options:
     1. Completely inaccurate
     2.
     3.
     4.
     5. Completely Accurate

10. - Question: Is information provided about tonight?
    - Options:
      1. Yes
      2. No

11. - Question: What weather information is given for tonight?
    - Options:
      1. Temperature
      2. Precipitation
      3. Cloud Cover
      4. Wind
      5. Thunderstorm
      6. Snow

12.
    - Question: How correct is the information about tonight on a scale of 1 to 5?
    - Options:
        1. Completely inaccurate
        2.
        3.
        4.
        5. Completely Accurate

13.
    - Question: Is information provided about tomorrow?
    - Options:
        1. Yes
        2. No

14.
    - Question: What weather information is given for tomorrow?
    - Options:
        1. Temperature
        2. Precipitation
        3. Cloud Cover
        4. Wind
        5. Thunderstorm
        6. Snow

15.
    - Question: How correct is the information about tomorrow on a scale of 1 to 5?
    - Options:
        1. Completely inaccurate
        2.
        3.
        4.
        5. Completely Accurate

16.
    - Question: Would you notice that this weather report was automatically generated if you read it on the Buienradar website? On a scale of 1-5
    - Options:
        1. It is very clearly auto-generated
        2.
        3.
        4.
        5. It looks 100% human written

17.
- Question: On a scale of 1-5, is the design of the weather forecast correct? In other words, does it start with this morning, then the evening and the night and end with the next day?
- Options:
  1. Not good at all
  2.
  3.
  4.
  5. Very good

18.
- Question: On a scale of 1-5 Do you think this is a good weather report, in terms of language and understandability, on a scale of 1 to 5?
- Options:
  1. Not understandable at all
  2.
  3.
  4.
  5. Perfectly understandable

19.
- Question: How satisfied would you be with this weather report, on a scale of 1 to 5?
- Options:
  1. Very unsatisfactory
  2.
  3.
  4.
  5. Highly satisfied

A.4 Sample AI-Generated Weather Report

Input provided by the user: Location - Maastricht; Date - July 19, 2023.

English Weather Report:

Overall, expect a day of mostly cloudy skies with occasional sunshine and a chance of rain. Temperatures will range from 14-24°C with winds blowing from the southwest and west. This morning in Maastricht, the temperature will be comfortable, ranging from 15-22°C with moderate cloud cover and occasional sunshine. There will be no precipitation or fog, and a light breeze will be blowing from the southwest. This afternoon, expect a moderate breeze from the southwest with temperatures ranging from 21 to 24 degrees Celsius. The day will be mostly cloudy with occasional sunshine and no precipitation. This evening in Maastricht, the temperature will be around 18-20°C with a moderate to high cloud cover and occasional rain showers possible. There is no indication of thunderstorms, snowfall, or fog. Tonight, the temperature will be mild, around 14-15°C with mostly cloudy skies and a light breeze blowing from the southwest or northwest. There will be no precipitation, fog, rain, snow, or thunderstorms expected. Tomorrow, expect a full day of mostly cloudy skies with occasional sunshine and a chance of rain throughout the day. Temperatures will range from 14-23°C with winds blowing from the southwest at a gentle breeze of 2 Beaufort.

Dutch Weather Report:

Vandaag in Maastricht kunnen we voornamelijk bewolking verwachten met af en toe wat zonneschijn en een kans op regen. De temperaturen zullen variëren van 14 tot 24 graden Celsius met een zachte wind uit het zuidwesten.
In de ochtend kunnen we een comfortabele temperatuur verwachten van rond de 15-22°C met een lichte bries uit het zuidwesten. De bewolking zal matig zijn met af en toe wat zonneschijn. Er wordt geen neerslag of mist verwacht.
Vanmiddag zal er een matige bries uit het zuidwesten waaien met temperaturen variërend van 21 tot 24 graden Celsius. De dag zal voornamelijk bewolkt zijn met af en toe wat zonneschijn en geen neerslag.
In de avond kunnen we een temperatuur verwachten van rond de 18-20°C met een zachte bries uit het westen en noordwesten. De bewolking zal matig tot hoog zijn met af en toe een regenbui mogelijk. Er is geen indicatie van onweer, sneeuwval of mist.
Vannacht zal het mild zijn met temperaturen rond de 14-15°C en een lichte bries uit het zuidwesten of noordwesten. De lucht zal voornamelijk bewolkt zijn met een bewolkingsgraad van 3-8 en er wordt geen neerslag, mist, regen, sneeuw of onweer verwacht.
Morgen kunnen we een volledige dag van voornamelijk bewolkte luchten verwachten met af en toe wat zonneschijn en een kans op regen gedurende de dag. De temperaturen zullen variëren van 14-23°C met een zachte bries van 2 Beaufort uit het zuidwesten.

A.5 Sample Raw Data

Eindhoven

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
1	#STN	YYYYMMDD	HH	DD	FH	FF	FX	T	T10N	TD	SQ	Q	DR	RH	P	VV	N	U	WW	IX	M	R	S	O	Y
2	370	20230101	1	220	110	120	190	163		59	0	0	0	-1	10087	82	8	49	23	7	0	1	0	0	0
3	370	20230101	2	210	100	90	180	156		57	0	0	0	0	10093	83	8	51		5	0	0	0	0	0
4	370	20230101	3	210	100	100	170	153		60	0	0	0	0	10095	82	8	53	2	7	0	0	0	0	0
5	370	20230101	4	210	100	100	150	151		65	0	0	0	0	10097	83	8	56		5	0	0	0	0	0
6	370	20230101	5	220	90	90	150	144		72	0	0	0	0	10104	83	8	61		5	0	0	0	0	0
7	370	20230101	6	220	80	80	150	141	137	75	0	0	0	0	10117	83	8	64	2	7	0	0	0	0	0
8	370	20230101	7	250	60	50	130	115		93	0	0	5	5	10131	62	8	86	81	7	0	1	0	0	0
9	370	20230101	8	230	50	40	80	109		99	0	0	10	11	10138	59	8	93	61	7	0	1	0	0	0
10	370	20230101	9	210	30	30	60	109		100	0	2	6	3	10147	62	8	94	81	7	0	1	0	0	0
11	370	20230101	10	190	30	20	50	106		96	0	15	0	-1	10149	65	8	93	23	7	0	1	0	0	0
12	370	20230101	11	220	50	70	90	129		93	0	32	0	0	10148	75	8	78		5	0	0	0	0	0
13	370	20230101	12	220	70	70	110	135	101	89	0	49	0	0	10151	75	8	73	2	7	0	0	0	0	0
14	370	20230101	13	220	70	70	110	136		80	2	45	0	0	10149	79	8	69		5	0	0	0	0	0
15	370	20230101	14	210	50	40	100	129		82	0	16	0	0	10148	75	8	73		5	0	0	0	0	0
16	370	20230101	15	180	30	30	50	130		84	0	5	0	0	10148	75	8	73		5	0	0	0	0	0
17	370	20230101	16	180	30	30	70	126		84	0	1	0	-1	10150	75	8	75	23	7	0	1	0	0	0

Figure A.2: Sample CSV Data from KNMI Weather Station in Eindhoven