DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Data Engineering and Analytics

# Computer Assisted Natural Language Description of Trends and Patterns in Time Series Data

**Siddhesh Kandarkar**

DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Data Engineering and Analytics

# Computer Assisted Natural Language Description of Trends and Patterns in Time Series Data

# Computergestützte Beschreibung von Tendenzen und Mustern in Zeitreihendaten in natürlicher Sprache


| | |
|---|---|
| Author: | Siddhesh Kandarkar |
| Supervisor: | Prof. Dr. Florian Matthes |
| Advisor: | Daniel Braun |
| Submission Date: | October 15th, 2020 |

I confirm that this master's thesis in data engineering and analytics is my own work and I have documented all sources and material used.

Munich, October 15th, 2020 Siddhesh Kandarkar

# Acknowledgements

I would like to first thank my supervisor Prof. Dr. Florian Matthes for giving me an opportunity to pursue this research.
I would like to thank my thesis advisors Daniel Braun and Marin Zec for their constant support and guidance. Thank you for keeping me motivated in tough times and for being patient with me throughout.
A big thanks to Celonis SE for this thesis opportunity. And thanks to all the Celonis associates who agreed to be part of my research work.
I am grateful to my family for their unconditional love, support, and belief in me. Thanks to all my friends who kept me motivated and assisted in my research work in some or the other way. Special thanks to my friends Manisha, Kasturi, Chirantar and Shweta for taking their time to proofread this report and help me improve it.

# Abstract

In today's digital world, data is of prime importance. In data-driven organizations, performance measurement based on analytical data has become an important business activity. In such organizations, key performance indicators (KPI) are often used in determining the performance of business processes. Time-series KPI data helps in tracking performances of KPIs over a period of time. Data experts collect, process and analyze such time-series KPI data sets to identify interesting business trends and patterns and explore hidden data insights, which eventually help the business stakeholders in making data-driven business decisions. Even with the availability of insightful analytical dashboards, business stakeholders often prefer intuitive and concise textual executive summaries for understanding business insights. The process of manually writing executive summaries can become tedious, repetitive and time-consuming for the data experts, especially when they have to write thousands of such reports on a regular basis. Thus, automating the process of writing executive summaries can prove beneficial for the data experts.

In this thesis, we analyze the manual and automated approaches for writing textual executive summaries. Next, we explore the different data-to-text systems which could be applicable for generating textual executive summaries for time-series KPI data. Further, we conduct task-based interviews with different types of users for identifying issues with the current process of manually writing executive summaries, and how it can be improved with the help of a software. Based on the answers and feedback from interviewees, we perform exploratory analysis for identifying user requirements for improving the process of writing textual executive summaries and further we identify an NLG approach that can be applied to our case study. Finally, we propose an automated reporting assistant prototype - RepGen, which can automatically generate textual executive summaries for time-series KPI data.

# Contents

# List of Figures

# List of Tables

# Part I.

# Introduction and Background Theory

# 1. Introduction

In today's digital world, data is ubiquitous and we are more dependent on technology than ever before. The advancement of technology is evident in our everyday life, with a huge amount of data being generated. The figure currently stands at 2.5 quintillion bytes per person, per day [1]. Having access to such a big repository of data opens up possibilities for research, analysis, and performance improvement. Data statistics from the year 2020 tell us that the majority of the world's data was generated only in the past two years [1]. Having data is not enough, we have to process and analyze it to extract hidden insights.

Data visualization tools like Tableau, Power BI, and Qlik Sense present large volumes of data in an understandable and coherent way in the form of visual elements like charts, graphs and histograms, which in turn help us comprehend the information and draw hidden insights from the data. In terms of business, data visualization helps business stakeholders analyze reports regarding sales, marketing strategies, and product interest. Based on the analysis, they can focus on certain aspects of their businesses which need more attention, and in turn, make more productive decisions.

Even with the data visualization tools and graphical representations, sometimes the data can be difficult to comprehend due to its sheer volume and complexity. There are situations where the person viewing the graphical representation might lack data interpretation knowledge, or might not have sufficient time to analyze all the visual representations and extract hidden insights from them. Sometimes, the graphical representations are themselves complex and difficult to understand. The most stunning visualization are futile until they convey the right message. That's where data experts re required, who can process and explore the data, and assist the business leaders in developing data-driven business strategies.

Data experts often convey important data insights in the form of a textual executive summary, thus combining data analysis with great storytelling. An executive summary conveys the key points of the report, and at the same time, includes sufficient information to help the reader make coherent and strategic business decisions. With a plethora of data being generated every day, business needs and requirements change on a daily basis. Thus, the role of a data expert becomes more crucial. But, the process of data analysis, interpretation and writing executive summary with dynamic data and ever-changing user requirements can become very tedious and time-consuming. The task of manually writing executive summaries for thousands of reports is repetitive, laborious and costly. Thus, an automatic reporting assistant tool can assist data experts in writing customized textual executive summaries for data-driven reports in a more efficient and timely manner.

## 1.1. Motivation

Celonis SE is an enterprise software company. It provides businesses with cloud-based, intelligent business solutions based on process mining technology, enabling them to visualize and improve the flow of their processes and thus transforming the way they work [2]. Celonis IBC uses the process mining software and consists of BI dashboards that provide different visualizations of business data i.e. time-series KPI) data in the form of process pipelines, graphs, plots etc. These visualizations help data experts and other stakeholders to observe and analyze the business processes in an effective manner.

However, representation of time-series KPI data in the form of visual representations is not always effective due to various factors such as complexity of the visual representations, difficulty in data interpretation, and excessive time required to understand complex visual representations. Thus, there are situations where data experts are assigned the task of writing executive summaries for time-series reports as per business needs and requirements. This process of writing executive summaries is often tedious, repetitive and time-consuming. Thus, automated generation of executive summaries can assist the data experts in writing executive summaries for time-series reports in a more efficient and timely manner. In Celonis, time-series KPI data is used for generating visual representations. Thus, for our thesis case study, we will be concentrating on the automated generation of textual executive summaries for time-series KPI data.

The objective of this master's thesis is to do a literature review on the current state-of-the-art techniques used in writing textual executive summaries for time-series data, for both manual and automated approaches. Further, we propose an automated reporting prototype to assist the data experts in writing intuitive and concise textual executive summaries for time-series KPI data.

## 1.2. Research Questions

The following research questions will be answered in this thesis:

1. What are the advantages and drawbacks with the current manual and automated

approaches for writing textual executive summaries for time-series data?

2. Which are state-of-the-art NLG approaches in the literature and practice for data-to-text generation of time-series data?

3. Can an automated reporting assistant tool prove beneficial for the data experts in writing textual executive summaries for time-series KPI data?

## 1.3. Approach

Thesis could be divided into two phases: conceptual approach and practical approach.

1. **Conceptual approach:**
    - Understanding and evaluation of the different approaches of manual report writing.
    - Understanding and evaluation of state-of-the-art automated reporting methods. Also, identification of automated reporting methods applicable to the thesis use-case of time-series KPI data.
    - Understanding different NLG approaches for data-to-text generation of time-series data.

2. **Practical approach:**
    - Requirement gathering and analysis: In our case-study, we gather requirements for an automated reporting assistant prototype from the expert (data analysts, BI analysts, consultants), and the non-expert (students, web developers, designers) users.
    - Based on the requirements, defining design goals, implementation tools and architecture for the prototype.
    - Develop an automated reporting assistant prototype.
    - Evaluation of the prototype with the end-users.

## 1.4. Thesis Structure

In the first chapter, we introduced the topic and motivation for the thesis. The remainder of the report is organized as follows. In Chapter 2, we introduce important terminologies like data-driven decision making, time-series data analysis, executive summaries, key performance indicators, and NLG. In Chapter 3, we introduce current approaches for manual and automated reporting for time-series data. In Chapter 4, we present state-of-the-art NLG approaches for data-to-text generation for time-series data. Here, we also give a brief overview of some of the commercial and open-source NLG tools for automated report generation. Chapters 5 and 6, gives a brief overview of the requirements engineering and software development process for implementing the prototype. Chapter 7 focuses on user feedback and evaluation attributes. Chapter 8 present the concluding remarks, limitations and future work.

# 2. Background Theory

This chapter introduces the concept of data-driven decision-making and states the key steps involved in making effective data-driven decisions. It also introduces key concepts like time-series data, KPI and executive summaries. Further, it introduces NLG, it's emergence and real-world applications.

## 2.1. Data Driven Decision Making

With access to a plethora of data, businesses are focused on exploiting that data for competitive advantages. Business experts knowledge on how to interpret and communicate data in an effective manner can help businesses in sound decision-making based on data-driven insights. Data-driven decision-making (DDDM) is about making business decisions based on analysis of data rather than just instincts and plain observations [3]. Incorporation of data in decision-making process depends on several factors, such as business goals, availability and quality of data, and expertise of the business professionals. Data-driven decision-making process can be summarized in the following steps.

1. **Understand the business objective:** Firstly, identifying and understanding thoroughly the problems in your given industry and competitive market. Identifying the business questions that need to be answered to achieve your organizational goals.

2. **Identify data sources:** Based on the business objective, identify the data-sources e.g. different databases, web-driven feedback forms, and social media content.

3. **Data cleaning:** Data cleaning is very important before data analysis is done. Data cleaning improves the quality of data by eliminating errors and inconsistencies from it.[4]. It helps in creating standardized and uniform data sets, which leads to more efficient data analysis. Data experts spend more time cleaning and organizing data, compared to the time they spend on analyzing data. This further emphasizes the importance of data cleaning.

4. **Analyzing data:** Once data collection, organization and cleaning are done, data analysis is performed on the input data. To analyze the data effectively, the data expert needs to have good knowledge of the data source, business domain and should know why the data is being analysed in the first place. Skill level required for data experts varies according to the complexity of the data and level of analysis required. In this step, it is also necessary to decide how to present the information to the business stakeholders.

5. **Turning insights into action:** The last step in data-driven decision making is coming to a conclusion. The conclusions drawn from the data analysis ultimately help organizations in making data-driven business decisions. Analysed data and insights are helpful only if they are presented compellingly to the business stakeholders. Thus, data experts must communicate their findings effectively with business stakeholders with good storytelling.

Time-series KPI data is one of the key elements used in Celonis IBC for creating visualization dashboards. Thus, our research is limited to written executive summaries of data-driven reports for time-series KPI data.

## 2.2. Time-series Data

Time-series data is a collection of observations over a period of time. Organizations can analyze important real-time and historical metrics with the help of time-series data. Time-series data can be displayed in dashboards with different types of visualizations such as line charts, tables and more.

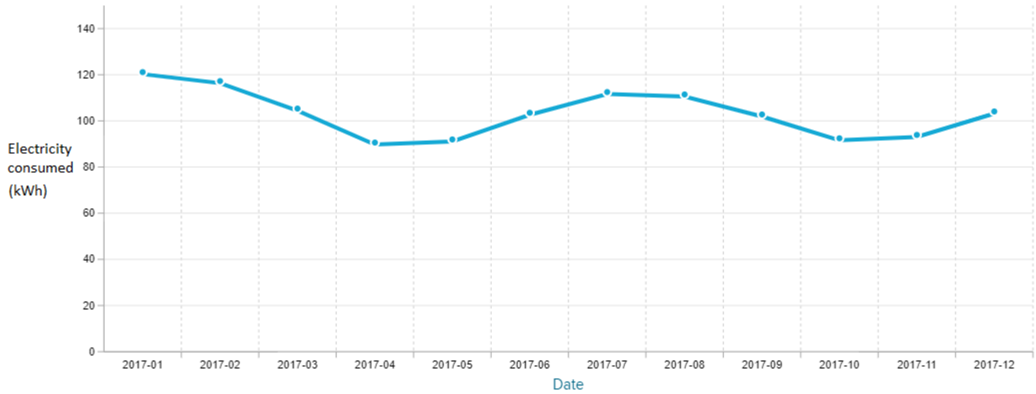



Figure 2.1.: Time-series chart example: Electricity consumption for the year 2017 [5]

A time-series chart or time-series graph is a data visualization tool used to graphically display time-series data. Each point on the chart corresponds to time/date and the value being measured at that point in time. In Figure 2.1, we have 'Date' as our x-axis and wind consumption in kilowatts as the y-axis, and each data point on the time-series chart corresponds to the amount of electricity consumption on the corresponding date. Good visual representations coupled with analytical tools help the organizations in analyzing the time-series data.

**Time-series data Analysis (TSA)** helps in understanding the time-series data by revealing hidden insights such as statistics, trends, patterns and overall changes in data over a period of time. Organizations use such analytical information to improve their existing processes and make strategic data-driven decisions. Time-series data analysis is widely used in many applications ranging from sales forecasting to process analytics. TSA can be used to detect interesting trends, patterns, seasonality, noise and other variations in data.

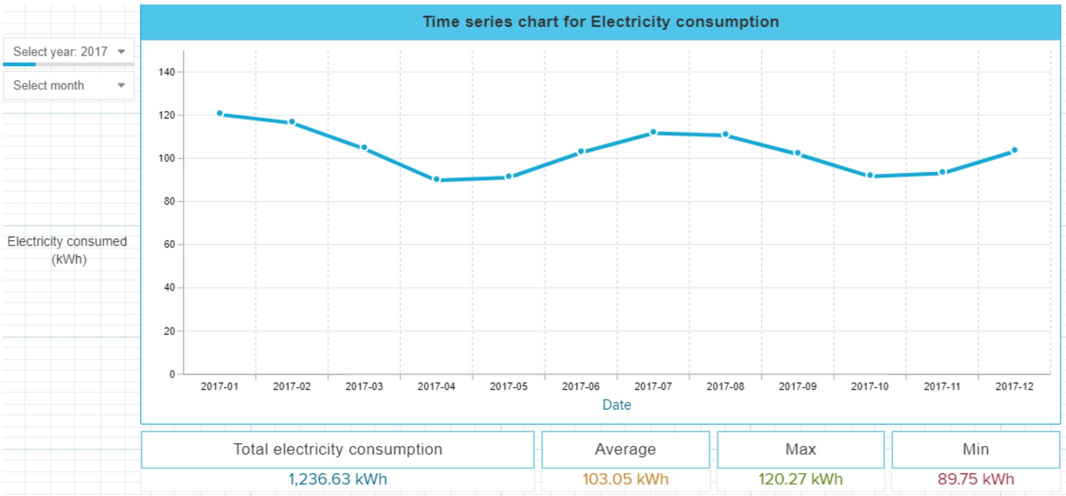



Figure 2.2.: Time-series dashboard: Electricity consumption in 2017 [5]

In Figure 2.2, we have a time-series dashboard for Electricity consumption. It consists of a time-series chart, along with related statistical information such as total electricity consumption in the year 2017, average consumption per month, maximum and minimum monthly consumption in 2017. We also have filters for year and month selection. Time-series dashboards can be augmented with more visual representations and statistical information depending on the requirement and complexity of the time-series data.

## 2.3. Executive Summary

An executive summary is an intuitive and concise textual summary of a business report. It restates the purpose of the report, highlights important points, and more importantly describes any conclusions or recommendations from the report. An executive summary is often written for business leaders, such as CEO's, department heads, or supervisors. A good executive summary can help them understand the critical information in quick time and make important business decisions. A time-series dashboard often consists of complex time-series charts and statistical data. Business users often find it difficult to process all the information available on a time-series dashboard. Also, the process is tedious and time-consuming, and they often end up requesting the data experts for executive summaries. One of the main objectives of our thesis is to propose an intuitive automated reporting assistant prototype, which should assist data analysts in writing intuitive and concise executive summaries.

## 2.4. Natural Language Generation (NLG)

In [6], Reiter and Dale have defined NLG as 'the sub-field of artificial intelligence and computational linguistics that is concerned with the construction of computer systems that can produce understandable texts in English or other human languages from some underlying non-linguistic representation of information'. Precisely defining NLG is rather difficult due to variations in input data such as numerical data, structured knowledge bases, and visual inputs such as images and videos [7]. With abundant data being generated and analysed on a daily basis, coupled with increasing dependency on data-driven reports for business decisions, organizations must find innovative ways to stay up. NLG could be the solution, especially for data-driven organizations. With NLG systems automating the data analysis and reporting tasks, data experts can focus on other important business activities. As shown in Figure 2.3, tabular data is transformed into natural language text with the help of NLG systems.

In NLG, the task of converting structured data into text can be divided into the following sub-tasks [9]:

1. **Content determination:** Deciding which information to include or exclude in the output text. This depends on the different filters and constraints applied to the input data.

2. **Text structuring:** Determining the order in which the data information will be presented in the output text. Good structuring can help the reader to understand and interpret the final text in a better way.

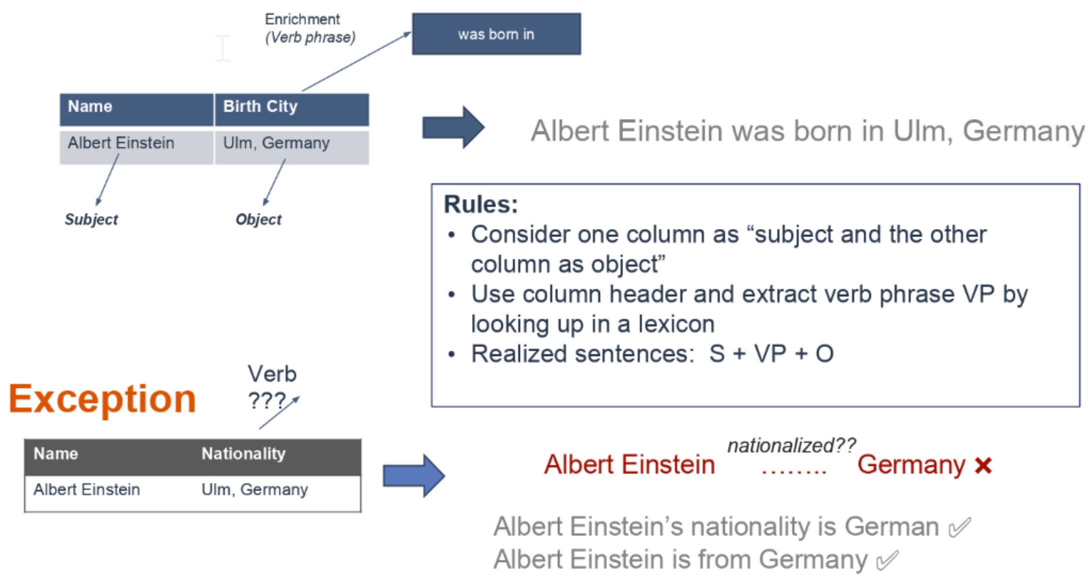



Figure 2.3.: Table description in Natural language text [8]

3. **Sentence aggregation:** Deciding which information to present in individual sentences. Proper aggregation of sentences can significantly enhance the fluency and readability of the output text.

4. **Lexicalization:** Selection of appropriate words and phrases to express the data information. Incorrect selection of words and phrases can lead to undesirable results. This is especially important when the output text is in multiple languages.

5. **Referring expression generation:** Selecting the words and phrases to identify domain objects.

6. **Linguistic realisation:** Combining all words and phrases into well-formed sentences. The objective is to produce a text which is syntactically, morphologically, and orthographically correct.

### 2.4.1. Architecture

The most common architecture in applied NLG systems is a three-stage pipeline architecture proposed by Reiter and Dale in [6]. The proposed architecture divides NLG into three stages:

- **Text Planning:** This stage combines content determination and discourse planning tasks.

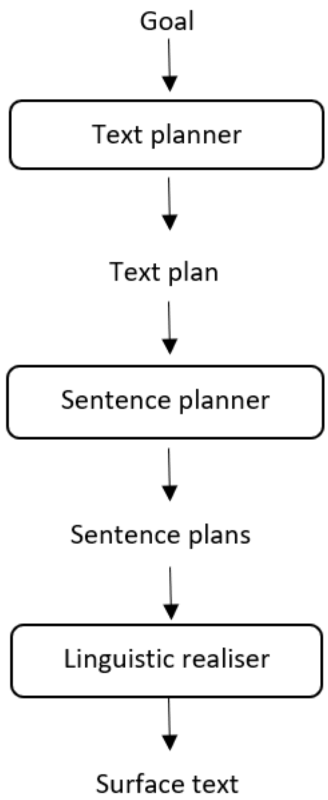



Figure 2.4.: Classical three-stage NLG architecture [6]

- **Sentence Planning:** This stage combines sentence aggregation, lexicalization, and referring expression generation.
- **Linguistic Realisation:** This task involves syntactic, morphological, and orthographic processing.

### 2.4.2. Applications

Following are some of the real-world NLG applications:

1. **Text summarization:** NLG systems can be used to generate intuitive and concise textual summaries based on the input data.
2. **Machine translation:** NLG text-to-text systems can be used for language translations.
3. **Dialog systems:** The system produces output texts based on the input queries. These systems are designed to simulate human-human conversation e.g chatbots. In recent

times, such systems have been used in conjunction with speech recognition and NLU to build voice-driven digital assistants [10]. E.g. Alexa, Siri, Cortana, Google assistant.

4. **Question answering (QA) systems:** QA systems are similar to dialog systems, with the only difference being that they do not preserve previous state i.e. there is no dialog history. Each input query is evaluated and processed individually. E.g. START QA system [11].

5. **Automatic text correction:** NLG text-to-text systems can be used for automatic grammar and spelling corrections.

6. **Report generation:** NLG data-to-text systems can be used to generate intuitive reports of databases and data sets. Such systems usually perform both data analysis and text generation. E.g. weather reports [12], soccer reports [13].

## 2.5. Textual Summary versus Graphical Presentations

In data-driven organizations, data experts communicate analytical insights to the business stakeholders in the form of visual presentations like charts and graphs, or textual summaries. The process of generating charts and graphs in visual dashboards is much faster and cheaper compared to the process of writing textual summaries. Also, the organization needs to hire data experts to write textual summaries and present it to the stakeholders. And even the data experts take considerable time and effort to write textual summaries. Even textual summaries have some advantages over graphical presentations. As discussed in chapter 1, sometimes the business stakeholders find it difficult to interpret and gain meaningful insights from graphical presentations due to their complexity and lack of clarity. There are situations where the stakeholders might lack expertise in understanding visual diagrams, or they might not have enough time to go through the complex charts and graphs, or there might be distribution constraints e.g. unable to view graphs and charts on smaller devices like mobile phones. Thus, there are situations where textual summaries are preferred over graphical presentations.

In the end, deciding which approach is more appropriate for given circumstance depends on a number of factors like the type of information being communicated, the amount of variation needed in the output text, the volume of reports to be generated, and most importantly on the needs and requirements of the business stakeholders. Thus, before developing a system for automatic generation of textual summaries, one must consider what benefits it offers over the conventional method of graphical presentations.

# Part II.

# Literature Review

# 3. Manual and Automated approaches for Writing Textual Executive Summaries for Time-series Data

Time-series data provides access to real-time and historical data. However, the usability of such data depends on its ease of access and understandability. The data is valuable only if it empowers business users to make key business decisions. Data experts such as data analysts and business analysts are responsible for data cleaning, processing and analyzing tasks. Even after having profound analytical data and key statistics, the end-goal is achieved only after a successful knowledge transfer of analytical data to the business leaders. This information is conveyed either with the help of business dashboards consisting of multiple visual diagrams and statistics or in the form of data-driven reports comprised of textual executive summaries. Compared to graphical representations, textual executive summaries are simpler, concise, more effective, and understandable [14]. Also, computer-generated texts can be superior to human-written texts [15]. Thus, business leaders often ask for executive summaries of data-driven reports, or a combination of visual representation and textual executive summaries. Textual executive summaries are either written manually by the experts or generated automatically using data-to-text generation tools. In this chapter, we are going to look into two approaches for writing textual summaries for time-series data: manual approach and automated approach.

*"The most important reason to include an executive summary is that in many cases, it is the only thing the reader will read"* says Pablo Bonjour, founder and CEO of Katy, Texas-based SMG Business Plans.

## 3.1. Manual Approach

Data experts like data analysts, business analysts, solution engineers, and business consultants who have profound knowledge about the data, businesses, clients, and more importantly have storytelling skills, are often assigned the task of writing executive summaries for data-driven reports. The difficulty level in the manual approach largely depends on the quantity and complexity of the time-series data sets that need to be analyzed.

### 3.1.1. Advantages

Manual approach of writing executive summaries for data-driven reports is beneficial when the data is easy to interpret and analyze, and the demand for such reports is very less. Following are some of the benefits.

- **Personalization:** Full control over what to write in the executive summary.
  E.g. what to include or exclude from the final summary, length of the executive summary.
- **Ease of use:** No external tool or software is required.
- **Ease of access:** It becomes possible to write and access textual summaries from any device.
- **Cost-effective:** There is no need to buy costly automated text generation tools. This is effective when the requirement for writing executive summaries is sporadic.

### 3.1.2. Drawbacks

Following are some of the drawbacks of the manual process of writing executive summaries, especially when the data is complex, and there is a high requirement for data-driven reports.

- **Tedious and repetitive:** The task of writing similar reports becomes monotonous and uninteresting over a period of time.
- **Time-consuming:** Writing hundreds of reports requires a lot of time.
- **Expensive:** Employees spend a lot of time writing hundreds of reports requested by different clients. This time can be utilized for more important tasks.
- **Static and inefficient:** Time-series data is very dynamic and changes every second. Whereas written executive summaries are static, and often contain old data. Thus, if a data report is not presented on time, it might not be that useful or effective for taking business decisions.

## 3.2. Automated approach

The manual process for writing textual executive summaries for time-series data is a two-step process. Firstly, the data expert has to analyze the data and extract important information and hidden insights from that data. Secondly, that information needs to be conveyed to the business leaders in an intuitive and effective manner, which helps to make strategic business decisions. But now the same steps can be embodied in software. Moreover, as mentioned in 3.1, the manual approach is not preferred for being tedious, repetitive, time-consuming, inefficient, expensive and static. Automation can make the task of writing textual summaries seamless by overcoming all the issues faced while writing textual summaries manually. With automation, time-series data can be translated into textual executive summaries in a timely manner.

### 3.2.1. Advantages of Automated Approach

- **Saves time:** Data experts spend an excessive amount of time in manually writing reports. With automated report writing, organizations can simply produce reports faster and with less effort.
- **Scalability:** Thousands of reports can be generated in a timely manner with minimal efforts.
- **Improve efficiency:** With automation, data experts can concentrate more on storytelling aspect rather than writing plain textual summaries.
- **Reduce risk:** Manually written reports are prone to statistical and grammatical errors. With automation, such errors can be avoided.
- **Dynamic and efficient:** Automated reports can be generated on button click, and can be sent to the business stakeholders on immediate request. Thus, the data-driven reports they receive are not outdated and can be helpful in making business decisions.

### 3.2.2. Hindrances in Automated Approach

Automating even a simple can be challenging. Here, we are talking about automating the entire process of writing textual summaries for data-driven reports. Many organization avoid implementing report automation mostly due to one of the following reasons.

- **Budget issues:** Commercial tools which provide automation feature are usually expensive.

- **Lacking expertise:** Buying automation tool is not enough. Employees having specific skill-set are needed to implement and configure such automation tools.
- **Integration issues:** Integrating external tools in existing systems can be challenging and time-consuming. Not all companies are willing to do so.
- **Trust issues:** Organizations are hesitant to share or upload their data on automated reporting platforms due to privacy and security concerns.

### 3.2.3. Data-to-Text Systems for Automated Report Generation

Data-to-text systems are used for automatically generating textual executive summaries for time-series data. They can be classified into two types: 1. Template-based systems, 2. NLG systems, and 3. Hybrid systems.

- **Template-based systems:** This is the simplest approach for data-to-text generation. These automated report generating systems follow a fill-in-the-gap approach, in which the system maps its input data directly to the output text. In template-based systems, the final textual documents i.e. data-driven reports generally consist of predefined templates with a number of gaps or blanks that need to be filled based on the dynamic input data. Such systems are easy to use and implement, and cost-effective. On the other hand, template-based systems lack flexibility and can be difficult to modify according to changing user needs. For example, if there are a huge number of templates, even a simple restructuring of sentence can be a lot of work. Thus, as explained in [6] template-based systems are useful only when limited syntactic variability is required in the output text.
- **NLG systems:** NLG systems are inspired by the architecture proposed by Reiter and Dale [6] which includes text planning, sentence planning, and linguistic realization. NLG system offer a number of advantages over template-based systems. One such advantage is maintainability, NLG systems are comparatively are easy to modify as per user requirements. Another advantage is the quality of generated output text, due to three different processing stages NLG systems can produce high-quality text. With sentence planning, NLG systems can perform aggregation over multiple related sentences. On the other hand, most of the NLG systems are complex, difficult to implement, and costly compared to the template-based systems. We have a detailed look at different NLG approaches in 4.

- **Hybrid systems:** Such systems use a combination of template-based and NLG techniques. NLG techniques can be used for high-level operations such as sentence planning, where as template-based techniques can be used for low-level realizations [6]. The decision on implementing such hybrid systems mainly depends on domain requirements and budget constraints.

# 4. State-of-the-art NLG approaches for Writing Executive Summaries for Time-series Data

Current computer systems present time-series data to human users either graphically by using visual representations or in a tabular form. In contrast, when a human presents time-series data to another human, he or she often uses language to do so. As mentioned in [14], graphical presentations in the form of graphs and charts work better for experts than for novices. However, when the graphical reports need more domain knowledge to explain, or needs to be communicated over low bandwidth devices such as mobile phones, textual summaries are more relevant. Good human-written summaries of data can certainly be effective, but they are expensive to produce due to time and cost required to do it manually for thousands of reports. [14]. The challenge for NLG is to produce executive textual summaries of time-series data automatically.

## 4.1. Previous Work

NLG systems always have textual data as the output with variations in the input data. For generating textual summaries for time-series data, NLG data-to-text systems are used. In this section, we will have a look at some of the previous work in the field of data-to-text systems for automatic report generation.

- **ANA** (1983) - Ana was the first application which used the technique of knowledge-based report generation for automatically writing stock reports. In this application, data from a Dow Jones stock quotes database served as the input, and the output text were the opening paragraphs of the stock market summary [16] .
- **TEMSIS** (Transnational Environmental Management Support and Information System)(1997) - This application was used to generate air quality report from environmental data. The environmental data used for report generation was a structured data which included measurement of pollutants, location details and time. The air quality can be

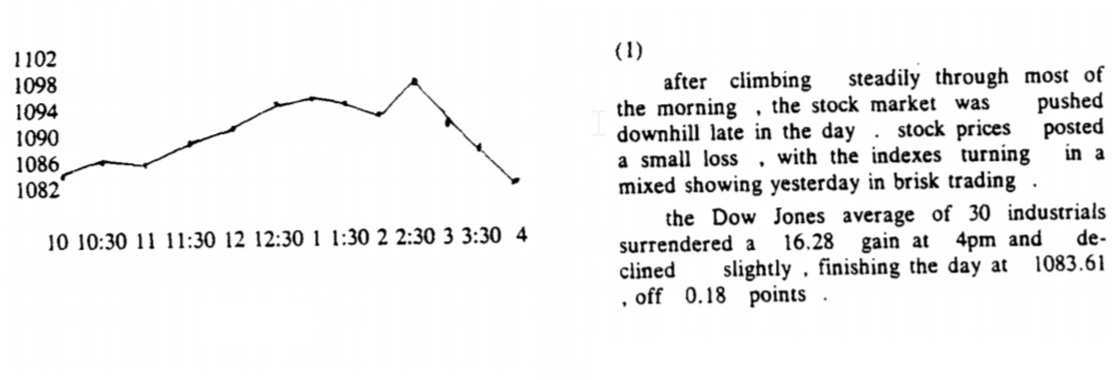



Figure 4.1.: ANA: report generated for time-series input data [16]

generated in German and French languages. These reports were editable, thus could be configured to fit additional details [17].

- **MultiMeo** (1998) - This application was used in the meteorology field to produce weather forecasts from structured data. It was an automatic multilingual short-report generation system. It automatically produced weather forecast reports in different languages like English, French, German, Spanish and Dutch [18]. Some of the other weather forecast generation systems were Forecast generator (FOG) [19] and Weathra [20].

- **TREND** (1998) - This application automatically generated summaries of historical time-series weather data. In TREND, edge detection techniques were used to detect significant trends in time-series data, which were then transformed into natural language summaries [21].

## 4.2. State-of-the-art NLG approaches:

Based on the underlying model, NLG systems can be divided into three major groups: rule-based, semi data-driven and fully data-driven. Following are the three NLG approaches:

1. **Rule-based approach:** This approach is an extension to the template-based approach 3.2.3, with user having the freedom to manipulate output text as per the requirements. It also provides more control to the users, where they can add conditional checks and grammatical rules as per their needs and requirements. In rule-based models, generation patterns are coded manually and are developed for restricted domains where the data input is limited and less complex. They use a limited vocabulary and a small set of

predefined syntactic structures. These models are robust, but lack flexibility and are too costly to develop and maintain. Predefined-text models, Template-based models and Grammar-based models follow the rule-based approach [22].

- **Predefined-text models:** It is the simplest NLG model in which the output is limited to a set of predefined output texts designed for the system. E.g. NPCEditor [23], an interactive dialogue system which answers user questions from a predefined pool of answers. It uses a text classification algorithm that maps user questions onto the predefined answers.
- **Template-based models:** As already explained in 3.2.3, these systems follow a fill-in-the-gap approach, in which the system maps its input data directly to the output templates. Templates contain variables which fill the empty slots based on the input data.
- **Grammar-based models:** These models use hand-crafted grammars and vocabulary to generate the output phrases. They can generate different templates using various vocabulary units. Thus, they are preferred over the template-based model, when a user needs variations in output text.

2. **Semi data-driven approach:** This approach adds statistical learning to the rule-based approach when the input data is numeric e.g. Time-series data. Generally, some statistical methods are applied to the input data for statistical analysis. This is done before applying rule-based constraints or sometimes even after that. In recent times with the availability of structure data, it has become easier to apply statistical methods.

3. **Machine learning-based approach:** It is a fully data-driven NLG approach. Different generation patterns for output text are learned from a training corpus in a supervised or unsupervised manner. In this approach, the main requirement is a large set of corpora to train machine learning models. The cost of preparing data and knowledge base is sometimes higher than the benefits gained by them. Compared to rule-based systems, the text quality of full data-driven systems is lower and the time required for development is comparatively higher as well. But, these systems are more adaptable and maintainable, and currently, there is a lot of research carried out in this field due to advances in artificial intelligence and deep learning fields. Also, rule-based systems are difficult to manage and implement when the data sets are huge and complex, extensive set of rules need to be manually written, or when a system needs to be implemented in multiple domains. Data-driven approaches may provide a solution for these shortcomings of the rule-based approach.

Following are some of the recent developments in natural language generation using data-driven techniques:

- **Markov chain:** In [24], Shannon proposed using Markov chain to create a statistical model of the sequences of letters. Markov chain was one of the first algorithms used in language generation. This model initially calculates the relationship between each unique word, and then for every input, it predicts the next word in the sentence by considering the relationship between each unique word to calculate the probability of the next word.
- **Recurrent neural network (RNN):** Neural networks are one of the methods used in artificial intelligence. As the name suggests, it tries to mimic the behaviour of the human brain. In RNN, each word is passed through a feedforward network in a sequential manner and the output of the current iteration is taken as an input to the next item in the sequence, allowing the information in the previous step to be stored [25]. RNN models can remember the background of the conversation, which helps them in language generation tasks. Such models are capable of capturing complicated languages and can be trained in an end-to-end fashion. However, RNNs are shortsighted [26] i.e. predictions are made based on only the most recent word, and they cannot store words that were encountered in earlier iterations. Thus, as the length of the sequence increases, RNNs are unable to produce coherent long sentences. E.g. Story Scrambler [26] is a RNN based test generation system.
- **Long short-term memory (LSTM) network:** LSTM was introduced to address RNNs inability to produce long coherent sentences. LSTM networks can selectively track relevant information and thus, remember information over a longer period of time. This makes LSTM networks ideal for time series analysis. But, LSTMs are often difficult to train due to their high computational needs and complex structure. E.g. Google Neural Machine Translation system [27] uses LSTMs to reduce translation errors.
- **Transformer:** Transformer is a deep learning model[28]. It uses attention mechanism to draw dependencies between input and output data. In Transformers, the sequential data need not be processed in any specific order. Thus, allowing more parallelization and reduction in training time compared to RNNs. E.g. Patent Claim Generation [29], where they have used OpenAI GPT-2 pre-trained model for generating patent claims.

## 4.3. NLG tools

NLG tools automate the process of language generation i.e. generating textual executive summaries. Automation helps organizations save valuable time and money. With the introduction of NLG tools, data experts no longer have to do the manual tasks of data interpretation, analysis and writing reports. Thus, helping organizations save valuable time and efforts. Moreover, NLG tools can scan across large volumes of data from multiple sources and generate reports automatically in minimal time and efforts. This is particularly valuable in the age of big data where a plethora of data is generated on a daily basis. With recent advances in the field of NLG, there are number of NLG tools available both commercial and open-source.

### 4.3.1. Commercial NLG Tools

With most of the organizations adopting data-driven approach, there has been a rise in the number of commercial NLG tools in recent times. AX Semantics and Arria NLG are two of the renowned NLG tools.

- **AX Semantics:** AX Semantics is a self-service NLG software. It follows the rule-based NLG approach, where users are allowed to configure their own set of rules and linguistic variances and elements in their preferred style to generate automated reports. Users are allowed to import and export their data and textual output in different formats including JSON, CSV and excel. It provides additional features such as auto-grammar correction, proofreading multi-language and cross-language generation.

- **Arria NLG:** It follows the rule-based NLG approach. Compared to AX Semantics, Arria NLG studio provides a more flexible and easy approach for rules creation and generating textual summaries with variations for a specific target audience. It offers on-premises and cloud deployment options.

  Apart from these, there are other commercial NLG tools like Wordsmith, Quill, SpeechKit, Alana and many more.

### 4.3.2. Open-source NLG Tools

Academics, students, and many companies prefer open-source tools and libraries for building NLG systems. Most of these libraries are used as linguistic realisation engines. Following are some of the widely used open-sources tools and libraries to build NLG systems.

- **SimpleNLG:** SimpleNLG is a library, written in Java, developed by Professor Ehud Reiter. It is capable of performing simple and useful tasks that are necessary for NLG [30] [31]. Since it is a library, we need to write our own Java program which makes use of SimpleNLG classes. SimpleNLG is intended to function as a "realisation engine" for NLG architectures, where we need to specify the words we want to appear in the output and their parts of speech. It basically creates a grammatically correct sentence from the parts of speech provided. It is an easy to use library, but it supports only English language at the moment.

- **RosaeNLG:** RosaeNLG is an open-source NLG library written in JavaScript, based on the Pug template engine [32]. It fully supports languages like English, French, German, Italian and Spanish, but you can generate texts in any other language with fewer features. It works in both node.js (server-side) and in the browser (client-side). It is fast, and complete enough to build real-life NLG projects.

- **NaturalOWL:** NaturalOWL is an open-source NLG engine written in Java. It produces texts describing individuals or classes of owl ontologies [33]. The ontologies are annotated with linguistic and user modeling annotations expressed in Resource Description Framework (RDF). An accompanying plug-in for the well known Proteg'e ontology editor is available, which can be used to create the linguistic and user modeling annotations while editing an ontology, as well as to generate previews of the resulting texts by invoking the generation engine. At present, NaturalOWN supports English and Greek. Compared to a simpler verbalize, NaturalOWL produces significantly better output text, and easy to use [34].

Part III.

# Methodology and Implementation

# 5. Requirements Engineering

Requirements engineering (RE) is an important milestone in a software development process. It involves defining, documenting, and maintaining requirements. In [35], Requirements Engineering Specialist Group (RESG) of the British Computer Society defines requirements engineering as "a key activity in the development of software systems and is concerned with the identification of the goals of stakeholders and their elaboration into precise statements of desired services and behaviour". Incomplete requirements, changing requirements and lack of user inputs are the three most challenging factors in software development [36]. Thus, an efficient requirement engineering process is utmost important in software development.

This chapter gives an overview of the requirements engineering process followed in the development of our automated reporting assistant prototype. It also states the methods used, challenges faced, and the outcome of the requirements elicitation and analysis phase.

## 5.1. Requirements Elicitation

Requirements elicitation involves researching and discovering the requirements of a system from the business stakeholders[37]. Most widely used requirements elicitation practices include interviewing the stakeholders, providing them with surveys or questionnaires, organizing workshops, conducting brainstorming sessions and prototyping. For requirement elicitation, we followed a task-based approach, wherein we completed the process in the following steps.

1. **Acquiring background knowledge:** It involves understanding the background and domain knowledge of the application that is being developed [38]. Since we were developing an automated data-to-text generation tool that would automatically generate textual executive summaries for input time series KPI data, it was important to have prior knowledge on concepts of time series KPI data used for process analytics in Celonis, available tools (Celonis snap) and dashboards, the current process of writing executive summaries for data-driven reports in Celonis, challenges with the current manual process of report writing, existing data-to-text translation systems, and existing NLG tools and approaches that adhere to our requirements.

2. **Identifying stakeholders:** Prior to application development, it was very important to know the target stakeholders. Stakeholders included clients, developers (who design, construct and maintain the system), and users (who interact with the system) [39]. For our case study, our main stakeholders were data experts who handled the tasks of writing executive summaries for data-driven reports. For an interactive application like RepGen, users are pivotal in the elicitation process. Users themselves are not homogeneous, and it was important to identify the needs of different user classes, such as novice users, expert users, occasional users, disabled users, and so on. For our case study, we conducted interviews with two different sets of stakeholders.
   - Experts: stakeholders having expertise and professional experience in writing executive summaries for data reports. It included data analysts, business analysts, consultants and solution engineers.
   - Non-experts: stakeholders having no previous experience related to writing executive summaries for data reports. It included software engineers, developers, consultants and students.
3. **Creating a time-series dashboard in Celonis snap:** We created an Electricity consumption analysis dashboard in Celonis snap. The data set used was Electricity consumption time series KPI data from 01 January 2017 to 01 August 2020. Interviewees i.e. the stakeholders were provided access to Electricity consumption dashboard in Celonis snap and assigned set of tasks.

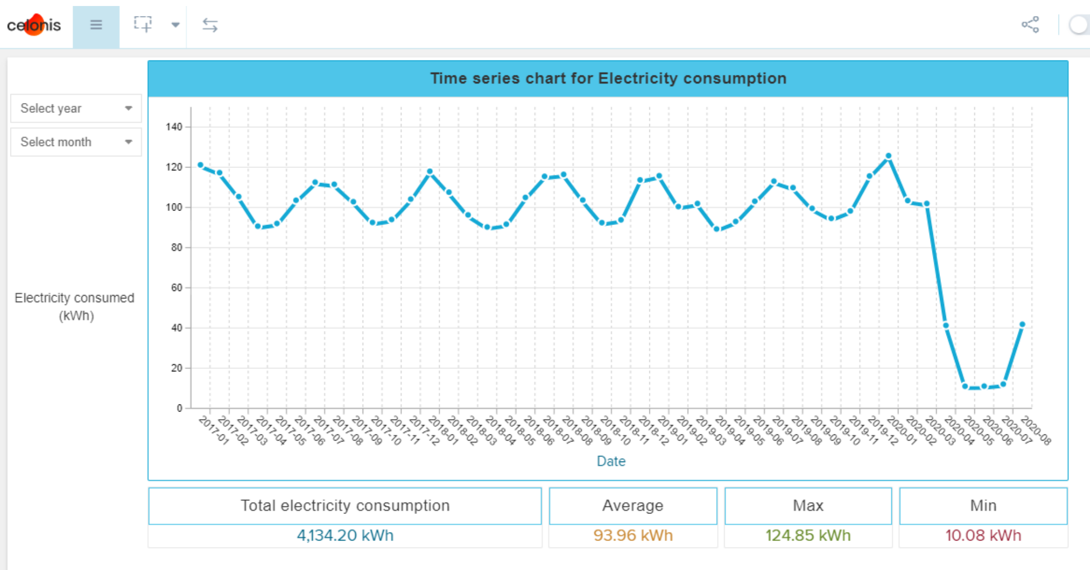



Figure 5.1.: Electricity consumption analysis dashboard [40]

4. **Conducting interviews:** Formal interviews were conducted with expert and non-expert stakeholders, as explained in Step 2. The interview was conducted in the following phases:

    • Providing problem statement and task-related information: Prior to the interview, interviewees were provided with a problem statement, and key information regarding tasks and data set. They were also provided with access to Celonis snap.

| | |
|---|---|
| Problem statement | RepGen GmbH wants to cut down on its annual electricity consumption due to budget issues. They have a meeting next week to discuss the same. Team manager Mr. Ramos wants some key information from you related to the electricity consumption in the form of a textual summary report. |
| Task-related information | We have created an Electricity consumption analysis dashboard in Celonis Snap. You will be assigned three reporting tasks related to the dashboard. You have to write a textual executive summary for each individual task. |
| Dataset information | Electric Consumption dataset: It includes monthly electricity usage data from 01-01-2017 to 01-08-2020 for RepGen GmbH. Each data point is captured on the 1st day of every month e.g. 1st Jan 2017, 1st Feb 2017 and so on. |

Table 5.1.: Problem statement, Task-related information and Data set information

    • Acquire demographic information: Demographic information helps in understanding the stakeholders and their classification. Thus, demographic information was collected from the interviewees such as name, gender, age, location, occupation, the position at work, organization name, work experience, prior experience in data analysis, experience in writing executive summaries, and writing data-driven reports.

    • Assign tasks: Three different tasks were assigned to the interviewees, and time required for completing each individual task was noted down. They were asked to write executive summaries for the following tasks 5.2:

    • Ask questions: A set of questions based on individual tasks were asked to the interviewees. The main objective was to find out difficulties faced while manually writing textual executive summaries such as:

        – Q1: Which part was more time consuming - data interpretation, deciding on

| | |
|---|---|
| TASK 1 | Write a textual summary for Electricity consumption in the year 2017. |
| TASK 2 | Compare Electricity consumption in 2017 and 2018. Write a textual summary for the same. |
| TASK 3 | Write a textual summary on unique/contrasting data observations for the entire time period (1st January 2017 to 1st August 2020). |

Table 5.2.: List of tasks assigned to the interviewees

what to write, or composing textual executive summary for the given task?

- Q2: What was more challenging– interpreting the data or writing a textual executive summary for the given task?
- Q3: Interviewees were asked to rate sub-tasks based on the difficulty level ( Figure 5.2).

| Sub-tasks | Strong disagree | Disagree | Neutral | Agree | Strongly agree |
|---|---|---|---|---|---|
| Understanding the Task was difficult | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ |
| Interpreting the data was difficult | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ |
| Deciding what to write was difficult | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ |
| Organizing the contents of the textual executive summary was difficult | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ |
| Writing a textual executive summary was more difficult than interpreting the data | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ |

Figure 5.2.: Difficult level for sub-tasks

- Q4: Suggestions to improve the current process of manual report writing.
- Q5: How could a software assist you in improving the process of report writing?
  How could a software assist you in making the process of report writing more efficient?

# 6. Prototype Design and Implementation

In the previous chapter 5, we performed the requirement engineering process by conducting task-based interviews with expert and non-experts in the report writing field. We also asked them a series to questions to better understand the issues related to the manual approach of writing textual summaries and how software can help them to overcome those issues. In Chapter 3 and 4, we surveyed state-of-the-art template-based and NLG approaches for data-to-text generation for time-series data.

Based on the results of the requirement analysis process, and evaluation of different automated data-text-generation approaches, we would like to propose an automated reporting assistant prototype named **RepGen**. As we have already stated in previous sections, the main objective of RepGen is to automatically generate highly customisable, accurate, intuitive and concise textual executive summaries for time-series KPI data. Thus, assisting the data experts in writing textual executive summaries for data-driven reports in a more efficient and timely manner. RepGen will be a single page web application with the functionality to automatically generate textual executive summaries for input time-series KPI data.

## 6.1. Design Goals

The first step to building any software application is to have well-defined design goals. Design goals help in concentrating on all the important components in our project and serve as a quality check by making sure the design meet the intended goals[41]. Thus, it is very crucial that we stick to the design goals in the process of application development.

It is important to consider the following points before starting with prototype implementation:

1. **End-users:** Before implementing a dashboard prototype, it is important to know the end-users, who are eventually going to use the prototype. In our case study for Celonis, we are developing an automated reporting assistant prototype for data experts such as data analysts, business analysts, and consultants who are directly involved in the process of report writing. Our prototype will assist them in writing executive summaries for business reports.

2. **Design requirements:**
    - Date selector: User should be able to manually select year, month, quarter as per requirement.
    - Event selector: User should be able to generate textual executive summary based on event selection.
    - Comparison selector: If the user wants to compare data analysis for different date periods e.g. different years, then the user should be able to do that.
    - Statistical data: Simple statistics such as total, minimum, maximum and average should be displayed on the dashboard.
    - Executive textual summary: Based on the selections, a textual executive summary should be generated, and displayed to the user. This text should be editable for the users, which can then be added to the final data report.
    - Data report: User should be able to add executive summary to the data report.
    - Download button: User should be able to download the final data report.

## 6.2. Business Workflow

In this section, we present a business workflow of the existing process of manually writing executive summaries for data-driven reports. The workflow steps were designed based on the observations from task-based interviews 5, in which the participants were assigned different tasks to write textual executive summaries for time-series KPI data. Further, we propose a novel business workflow of our prototype.

### 6.2.1. Manual approach

As already discussed in chapter 3, manual approach for writing textual summaries works well only when the data-set is small and simple to interpret, and the requests for writing executive summaries for data-driven reports are less frequent. When either of the conditions fail i.e. when the data is complex or there is a high requirement for writing executive summaries, manual approach can prove tedious, repetitive, time-consuming, and inefficient.

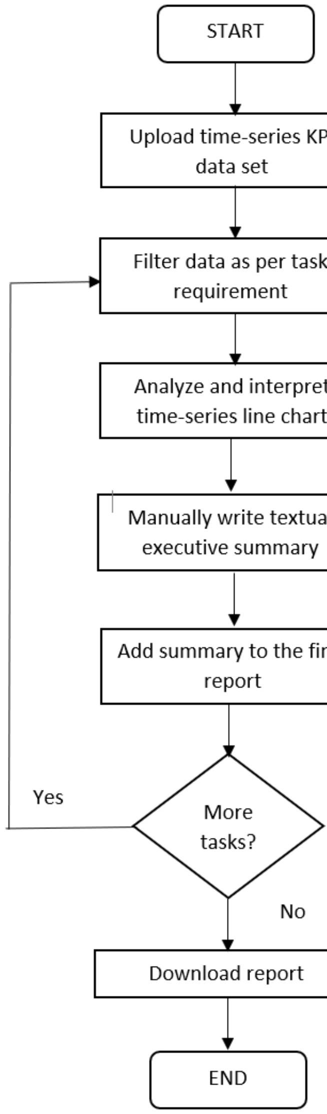



Figure 6.1.: Workflow flow diagram: Manual approach of writing textual summaries

In workflow diagram 6.1, the data experts will spend most of their time in manually writing executive summaries, and this process becomes tedious, time-consuming and repetitive over a period of time with more number of tasks. To overcome these issues, we propose an alternate business workflow with RepGen.

### 6.2.2. Automated approach using RepGen

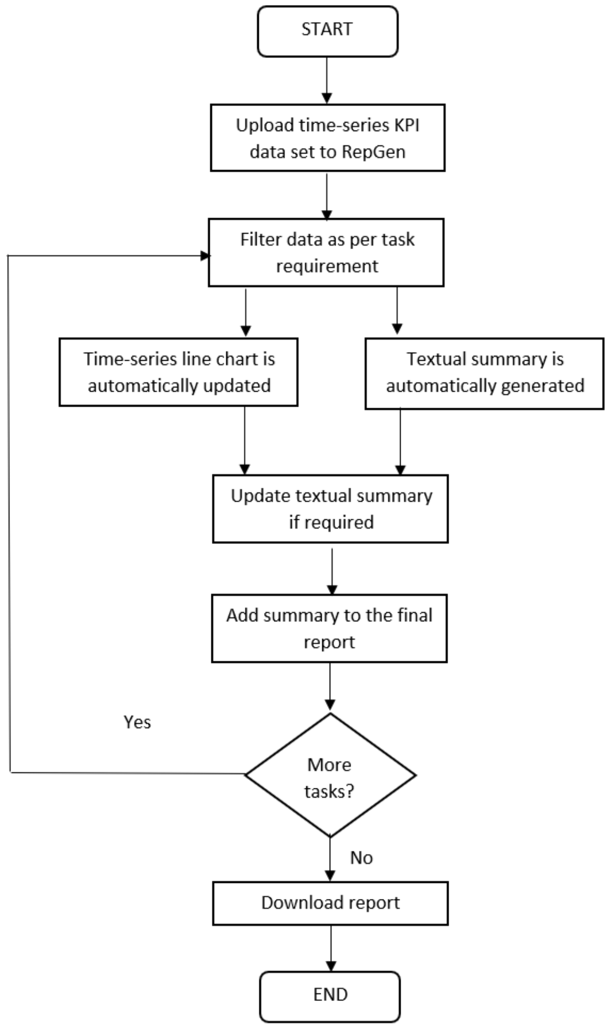



Figure 6.2.: Workflow flow diagram: an automated approach using RepGen

In workflow diagram 6.2, there are few changes in the iterative portion. The time-intensive task of writing textual executive summaries is now automated and taken care of by the application. The data expert still has the choice to update the executive summary if required, but this is comparatively much better than the manual approach. Now, data experts can concentrate more on understanding and analyzing the data (by using different selection), rather than writing.

## 6.3. Prototype Implementation

One of the main objectives of this thesis is to propose an automated reporting assistant prototype. In this section, we explain what is prototyping, and the need for implementing prototype before the actual product development. Further, we explain technical details related to the prototype.

### 6.3.1. Prototyping

A prototype is an exact representation of the actual application, that might be developed in future. It provides a general overview of the application functionalities and system workflow. It is generally presented as a proof-of-concept to higher authorities in an organization and is useful in getting user feedback before actually developing the application. Requirements elicitation helps in gathering the requirements of the end-users, but it is easier to discover their expectations and additional requirements by providing them with a model that resembles the real product in many ways. Following are some of the benefits and drawbacks of building prototypes.

**Benefits:**

- Serves as a proof-of-concept
- Reduces the cost and time of development.
- Helps in gathering additional requirements that were missed in requirements gathering process.
- Provides user satisfaction.

**Drawbacks:**

- The end-users may expect the finished product to be exactly same as the prototype. It might not always be possible to due technical and practical issues.
- Developers may be tempted to stop with the prototype, especially when it satisfies all the user requirements.

### 6.3.2. Development Framework

RepGen is developed in Dash framework. Dash is a python framework for building interactive analytical applications. It is built on top of Plotly.js, React and Flask. This adheres to our application requirements and design goals.

### 6.3.3. Data Format

NLG systems can have different types of inputs, but the output is always in textual format i.e. human language. Thus, prior to the application development, it is important to answer two important questions:

1. What will be the input data?
    - In our prototype, our input data will be time-series KPI data. The input will be time-series KPI data table, consisting of date-time values as the primary column and KPI values as the secondary columns. There can be a single KPI or multiple KPIs in a single data table. User can upload the data in excel format.
2. What will be the structure of the output text?
   The output text will always be in plain English. It can be decomposed into three sections [6]:
    - Unchanging text: Textual content that is always present in the final report.
      E.g. report title with company name, date of report generation, footer text.
    - Directly available text: Textual content that is directly available from the input data.
      E.g. table name, KPI name.
    - Computational data: Textual content based on analytical data.
      E.g. For electricity consumption KPI time-series data, it will be total electricity consumption for a particular year, average consumption.

## 6.4. User Interface

In this section, we present the user interface of our automated reporting assistant prototype - RepGen. It is a single page application containing several components. We will have a look at all the component, their functionality and navigation.

1. **Home page:** RepGen is a single page application, thus has only one page. It comprises of different components, which will be discussed later.

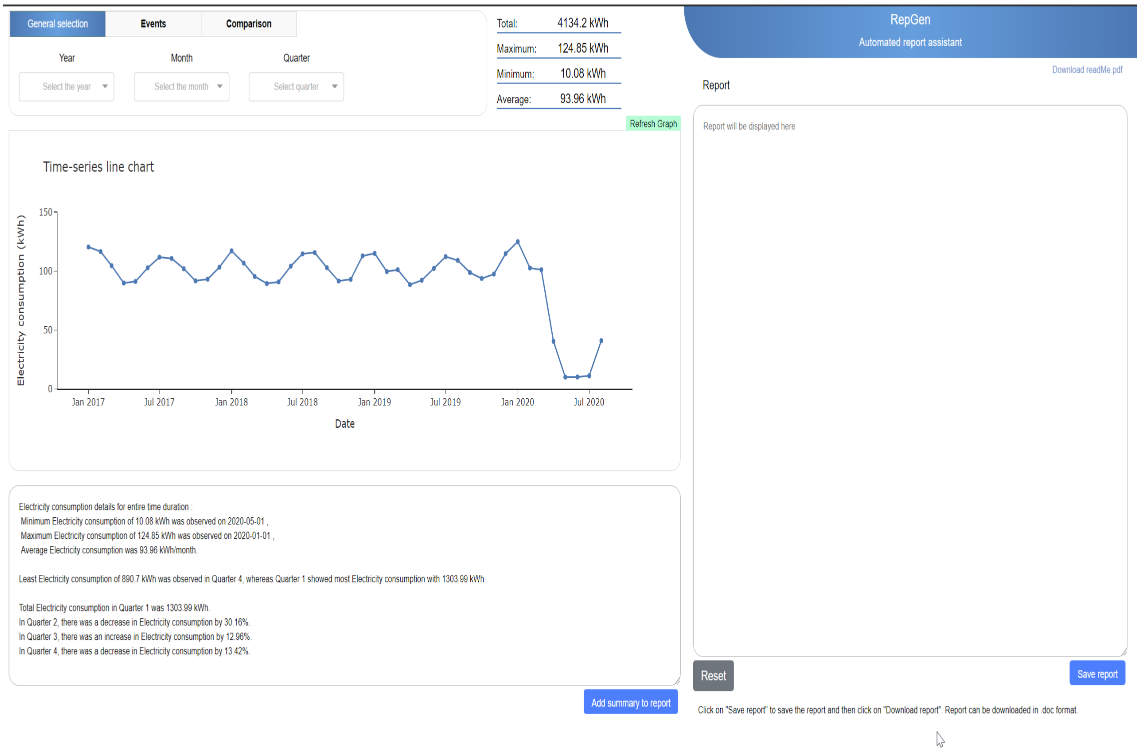



Figure 6.3.: RepGen: Home page

2. **Selection components:** It consists of three different tabs, which can be used for different data selections by filtering the data. It is located at the top left section of the home page.
   - **Tab 1 (General selection):** It consists of basic configuration settings. Line-chart and textual executive summary will be dynamically updated on a particular selection or combination of selections. Here, we have provided three drop-downs for year, month and quarter selection.

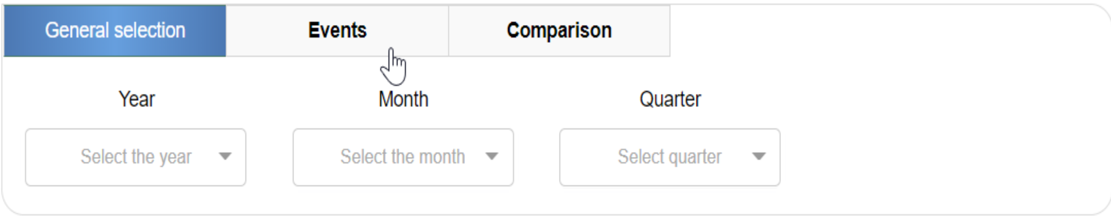



Figure 6.4.: Tab 1: General selection

- **Tab 2 (Events):** User can select from a list of events. Line-chart and textual executive summary will be dynamically updated on event selection.


Figure 6.5.: Tab 2: Events

- **Tab 3 (Comparison):** User can use this tab to generate executive summaries based on yearly comparison.

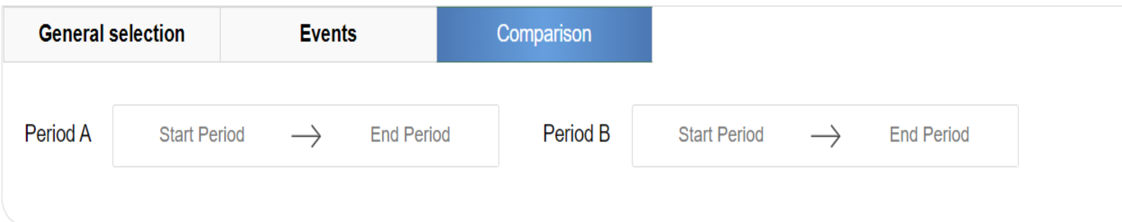



Figure 6.6.: Tab 3: Comparison

3. **Time-series line chart:** A time-series line chart is dynamically generated for the input time-series KPI data. It automatically updates with every selection. Line charts are helpful in visualizing trends and patterns in data over a period of time. It helps data experts in interpreting the automatically generated textual summary.

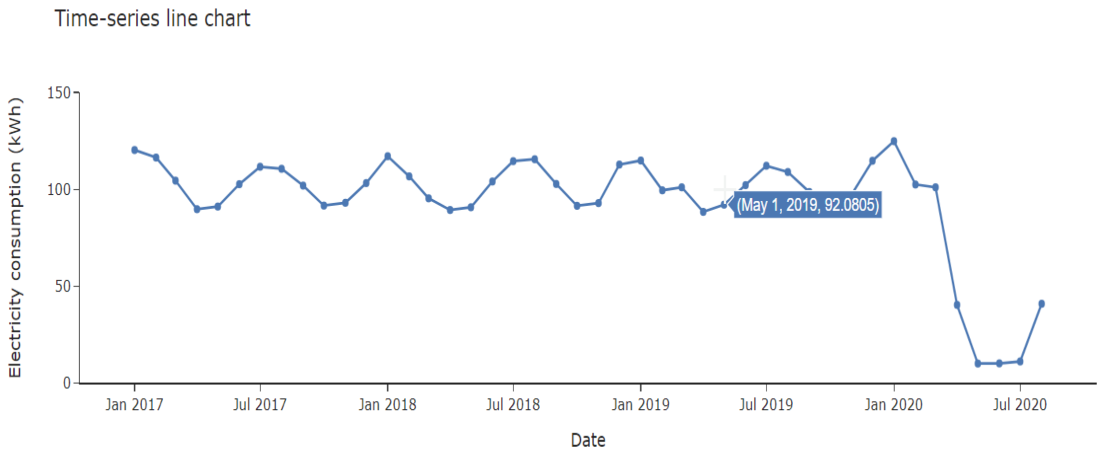



Figure 6.7.: RepGen: Time-series line chart

4. **Statistics:** The statistical data for the time-series KPI data is dynamically generated in Statistics component. We have four variables - Total, Maximum, Minimum and Average. The values for these variables automatically change with every selection.

| | |
|---|---|
| Total: | 4134.2 kWh |
| Maximum: | 124.85 kWh |
| Minimum: | 10.08 kWh |
| Average: | 93.96 kWh |

Figure 6.8.: RepGen: Statistics

5. **Textual summary:** This is the most important component of this application. A textual summary is generated based on the current data selection. By default, it displays a textual summary for the entire duration. Similar to time-series line chart and statistics components, textual summary is also dynamically updated with every selection. The generated textual summary is editable, thus data experts can update the text as per their needs and requirements. We have a 'Add summary to report' button which can be used to add a summary for the current data selection to the final report.

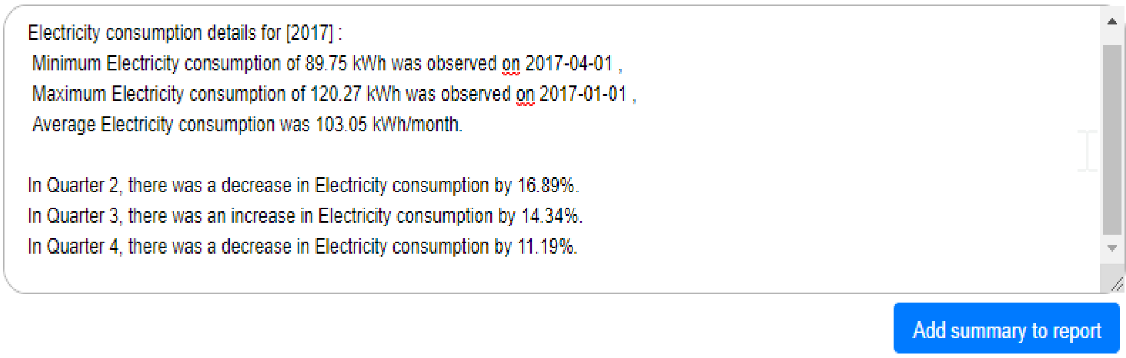



Figure 6.9.: RepGen: Textual summary

6. **Report section:** With multiple data selections, the user can add a corresponding textual summary to the final data report. Once all the tasks are complete, the user can download the report.

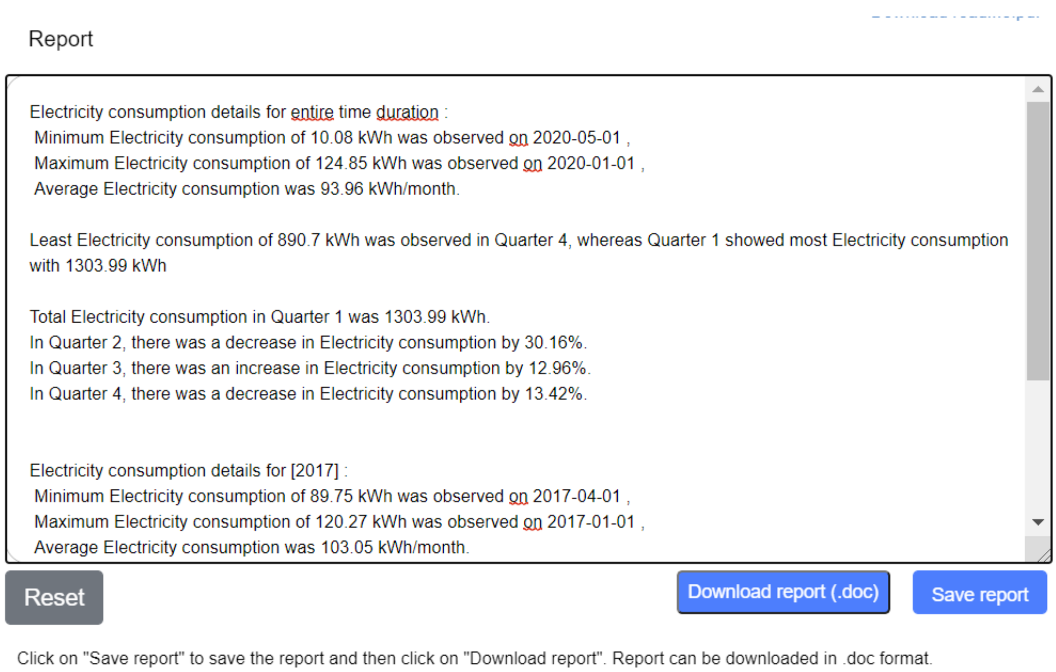



Figure 6.10.: RepGen: Final report

Here we have 3 buttons:

a) Reset button - to delete all the current report text.

b) Save button - to save the current state of the report

c) Download button - to download the report. At present, the report can be downloaded in .doc format.

# Part IV.

# Feedback and Evaluation

# 7. Evaluation

Evaluation is an important process in web application development. And it becomes more important in case of systems dealing with Natural Language Generation (NLG) because such systems are difficult to evaluate due to the subjective nature in evaluation of human languages. In most of the situations, there is no good or bad output text, every user might perceive it in a different manner depending on his business knowledge, grammatical knowledge, personality and much more. Some NLG tasks are even domain-dependent [42]. Thus, it is very difficult to evaluate NLG based applications.

## 7.1. Evaluation Techniques

Traditional NLG evaluations can be classified into two types: intrinsic or extrinsic [43].

1. **Intrinsic evaluation:** Intrinsic evaluations of NLG systems seek to evaluate the properties of the system. This evaluation technique can be further classified into two types:
   - **Human evaluation / Human ratings:** Human-based evaluation has always been the most popular and preferred way of evaluating NLG systems. In this type of evaluation, subjects are provided with both NLG and human-written texts, and asked to rate both the texts. And based on this rating, NLG systems are evaluated. Sometimes, subjects are asked to rate the text generated by NLG systems based on different metrics like fluency, accuracy, diversity etc.
   - **Corpus-based evaluation:** NLG systems are also evaluated by comparing the text generated by the NLG system to the corpus text through the use of automated metrics like BLUE and ROGUE [43].
2. **Extrinsic evaluation:** This a task-based evaluation technique. The subject is provided with text generated by the NLG system, and then the effectiveness of the system is measured on how well the generated text helped the subject to perform a given task.

With corpus-based evaluations, one can perform repeatable evaluations for a NLG system. But, it requires aligned corpora which are not always available [44]. On the other hand, task-based evaluations are time-consuming and expensive and can be difficult to carry out. Whereas, human-based evaluation technique is easy to implement and produce better results compared to other techniques [44]. Thus, we will be using human-based evaluation technique to evaluate our prototype.

## 7.2. Prototype Evaluation

The ultimate goal is to identify how useful RepGen is at assisting the data experts and other stakeholders in writing textual executive summaries for data-driven reports.

### 7.2.1. Method

To carry out a human-based evaluation on our prototype, an online survey was conducted and the participants were asked to answer a short questionnaire. We also provided online access to our prototype - RepGen, and asked the participants to use the application and provide feedback accordingly. To make the process easier for evaluation, we used the same Electricity consumption data set in the prototype, which was used during task-based interviews.

### 7.2.2. Participants

The online survey was sent out to only those participants who were involved in the requirement gathering process. Thus, the number of participants in the online survey were limited. The participants were a mix of experts and non-experts in the field of report writing. It had a good mix of participants from both the genders, as well as a good mix of business professionals and students.

### 7.2.3. Evaluation metrics

Human evaluation of our prototype RepGen was done in terms of following metrics:

1. **Intuitiveness:** It checks whether the generated text was easy to understand and interpret.

2. **Quality:** It checks whether the quality of the generated text is as per business requirements and standards.

3. **Accuracy:** It checks whether the details present in the output text are accurate and precise.

4. **Fluency:** It checks how well-formed and grammatically correct the output text is.

### 7.2.4. Questions

The online survey had the following set of questions:

1. User expertise: Participants were asked to rate themselves from 1 to 3 in terms of their expertise in writing textual reports. (1 - non-expert, 3 - expert)

2. Ease of report creation: How easy was it to generate a report using RepGen? (1 - very difficult, 5 - very easy)

   Questions 3-7 were based on evaluation attributes for textual summary -

3. Intuitiveness - How easy was it to understand and interpret the automatically generated textual summary? (1 - very difficult, 5 - very easy)

4. Quality - How was the quality of the generated textual summary? (1 - worst, 5 - excellent)

5. Accuracy - How accurate were the details presented in the textual summary? (1 - inaccurate, 5 - accurate)

6. Fluency - How well-formed and grammatically correct the textual summary was? (1 - worst, 5 - excellent)

7. Overall impression - Participants were asked to rate the prototype on the basis of its overall impression. (1 - worst, 5 - excellent)

### 7.2.5. Results

In this section, we will have a look at some of the interesting results that were observed from the results of the online survey-

1. **Diversity in participants:** Diversity among the participants is very important to have unbiased results from the survey. The survey was carried out with 10 participants. There was diversity among the participants in terms of the level of their expertise in report writing, gender, and occupation.

    - Expertise - Participants were asked to rate themselves from 1 to 3 in terms of expertise they have in writing report. (1 - non-expert, 3 - expert)

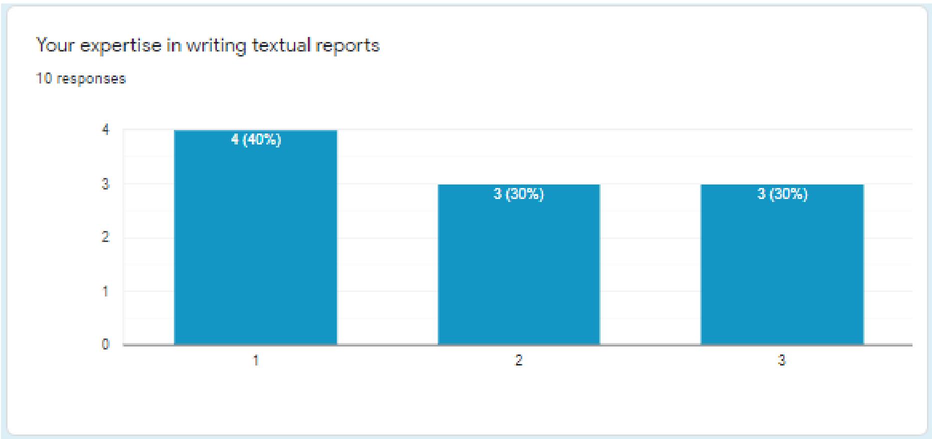



Figure 7.1.: Survey analysis: Level of expertise

    - Gender - There were 5 male and 5 female participants who were part of the online survey.


Figure 7.2.: Gender-based classification

- Occupation - Participants were from different fields of studies and work took part in the survey.

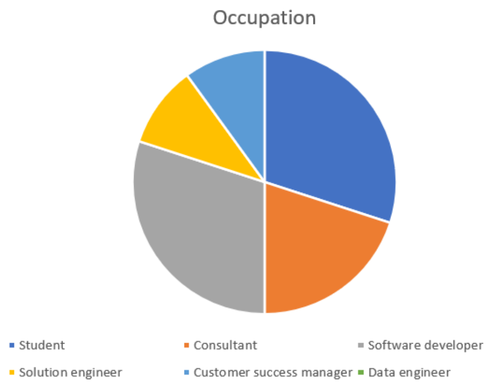



Figure 7.3.: Occupation-based classification

From the results, we can conclude that there was good diversity in the participants in terms of expertise, gender parity and occupation. This helps in generating unbiased results from the survey.

2. **Evaluation metrics** The next set of questions[Q3 - Q6] concentrate on the evaluation of automatically generated textual summaries on the basis of 4 text evaluation metrics - Intuitiveness, Quality, Accuracy and Fluency. Participants were asked to rate their evaluation on the Likert-scale from 1 to 5. E.g. For Accuracy - participants were asked how accurate were the details presented in the textual summary? (1 - inaccurate, 5 - accurate)

   - Intuitiveness: Understanding and interpretation of the automatically generated textual summary.

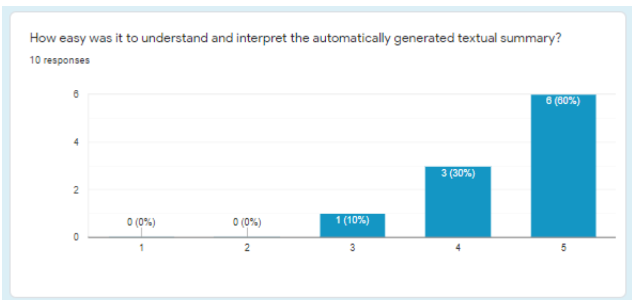



Figure 7.4.: Survey analysis : Intuitiveness of textual summary

   - Quality: Quality in terms of words used, sentence formation etc.

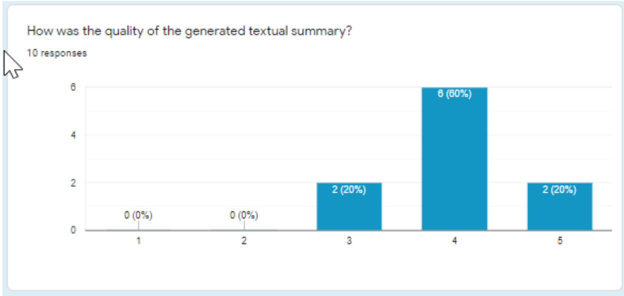



Figure 7.5.: Survey analysis: Quality of textual summary

- Accuracy: In terms of details presented in the textual summary. E.g. percentage increase and decrease, maximum value, minimum value, average value etc.

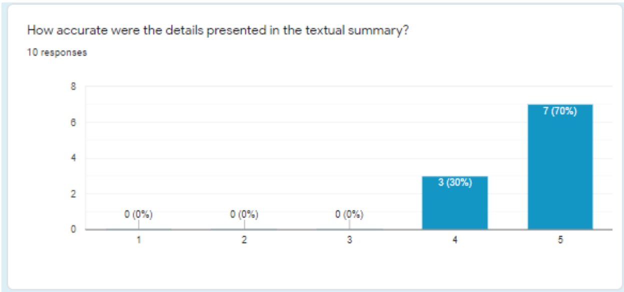



Figure 7.6.: Survey analysis: Accuracy of textual summary

- Fluency: It tells how well-formed and grammatically correct the textual summaries were.

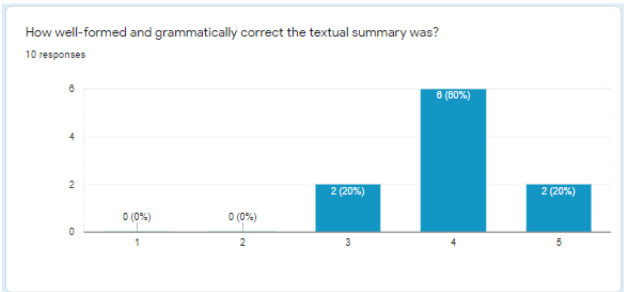



Figure 7.7.: Survey analysis: Fluency in the textual summary

From the above results we can observe that for all the 4 metrics, the distribution of ratings was from 3 to 5 i.e. from average to excellent. We can also conclude that the participants found the generated textual summaries very intuitive [average rating = 4.5] and very accurate as well [average rating = 4.7]. For both quality and fluency, the average rating was 4, i.e. between neutral and excellent. Thus, we can conclude that automatically generated textual summaries were intuitive and accurate, but improvements can be made in terms of quality and fluency.

3. **Overall impression of the prototype:** The average rating for this question was 4.4, with 40% rating 5,and 60% rating 4 on the Likert scale. Thus, we can conclude that the overall feedback for the implemented prototype was very positive and encouraging.

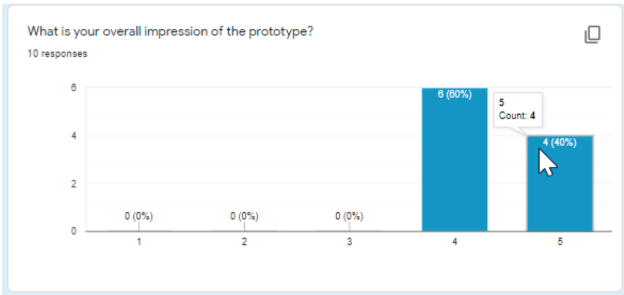



Figure 7.8.: Survey analysis: Overall impression of the prototype

4. **Suggestions for improvement:** Participants provided some interesting suggestions that would help in the betterment of the prototype. There were suggestions to improve the quality and structure of the textual summary, to include tables within the textual summary for comparison tasks, and also to include a text editor to customize text styling.

# Part V.

# Conclusion and Future work

# 8. Conclusions and Future Work

This chapter summarizes the work presented in the thesis. We discuss whether the research questions were answered and thesis objectives were achieved through literature review and prototype implementation. Next, we look at the limitations of our current thesis work. And in the end, we discuss future work which can help us to overcome the limitations.

## 8.1. Findings

The objective of this thesis was to analyze the current approaches in writing textual executive summaries for time-series KPI data and propose a prototype that can automate this process and assist the data experts in writing textual executive summaries in a more efficient and timely manner. From the results, we can conclude that our proposed prototype - RepGen was able to automatically generate textual summaries for time-series KPI data. These textual summaries were customizable and adhered to good extend to the NLG text evaluation metrics i.e. accuracy, intuitiveness, fluency and quality of text. In the remainder of the section, I have restated the research questions and provided my findings for the same.

**RQ1 - What are the advantages and drawbacks with the current manual and automated approaches for writing textual executive summaries for time-series data?**
Firstly, we analyzed the advantages and drawbacks of both the manual and automated approaches 3. From the analysis, it can be concluded that an automated approach was more efficient and convenient for data experts, and can be recommended especially when there is a huge requirement for writing executive summaries. Considerable cost and effort is involved in building data-to-text automated systems. Thus, if the requirement for writing executive summaries is sparse and if the organization has issues like the budget problem, lack of expertise, integration issues etc., then the manual approach is recommended.

**RQ2 - Which are state-of-the-art NLG approaches in the literature and practice for data-to-text generation of time-series data?**

The main objective of the thesis was to automate the process of manual report writing for time-series KPI data. Data-to-text systems are used for generating natural language for time-series data. In this thesis, we reviewed state-of-the-art data-to-text NLG systems. Chapters 3 and 4 give an overview of the different approaches used in data-to-text generation. The nature of data-to-text systems made it difficult to directly compare different NLG systems even with the same environmental setup. As presented in Chapter 5, task-based interviews were carried out with different sets of participants having varying levels of expertise in writing data-driven reports. Evaluating the issues with the current manual approach of writing executive summaries, and gathering user requirements, especially from data experts from Celonis, we identified the NLG approach that is best suited for our prototype. We decided to use rule-based NLG approach. As discussed in Chapter 4, even though machine learning based NLG approaches are more maintainable and flexible to textual variations, the text quality is lower and the development time is higher compared to template-based and rule-based approaches. Considering other factors like the complexity of time-series data and the limited time available for developing a prototype, it was decided to use rule-based NLG approach for developing a reporting assistant prototype.

**RQ3 - Can an automated reporting assistant tool prove beneficial for the data experts in writing textual executive summaries for time-series KPI data?**

As explained in Chapter 7, the evaluation of natural language generating systems is a difficult task. In this thesis, we have presented an automated reporting assistant prototype - RepGen. It is a rule-based NLG system, that automatically generates textual executive summaries for time-series KPI data. The prototype is implemented in Dash framework, with Python as the programming language. It is a single page web application, comprising of different selection components and automated features such as the automated generation of textual summaries, dynamic visualization updates, and dynamic updates of statistical data. Our online survey results with both the experts and non–experts demonstrate that the prototype generates textual summaries which adhere to good extend to the NLG text evaluation metrics. The metrics used for evaluation were accuracy, intuitiveness, fluency and quality. The feedback received from the survey participants was positive and encouraging. The suggestions can be further used to improve the prototype. Overall, RepGen is a good step forward in developing an enterprise-level rule-based NLG application for generating automated textual executive summaries.

## 8.2. Limitations and Future Work

In this thesis, due to limited time, we could conduct task-based interviews with only a small set of participants. More number of interviews, especially with data experts would have helped in an in-depth analysis of the issues concerned with manual report writing and would have helped in defining more conclusive user requirements. Also, with limited time for development and testing of the prototype, we had to limit ourselves to implementing only a subset of user requirements.

There is no conclusive evidence provided that rule-based NLG approach is the best approach for automated generation of textual summaries for time-series KPI data. Implementing similar prototype with other approaches such as template-based and machine learning-based, and then comparing their performances and results would be interesting.

As the prototype we developed can be potentially used at the enterprise level. Following are some of the features which can be implemented in future.

- **Realisation engine:** Linguistic realisation engine coupled with the rule-based approach would be helpful in generating more fluent textual summaries. It would be possible to add more variations to the output text as well.

- **Output in multiple languages:** RepGen is limited to English language at the moment. Multiple language output can be beneficial.

- **Multiple options:** In RepGen, there is only a single output summary generated for every input data selection. There was an user requirement for multiple options of textual summaries to choose from. Implementation of realisation engines can provide users with an option to add textual summary from multiple output suggestions.

- **Improved trends and pattern detection:** RepGen can detect trends by applying rule-based conditions on the input data. It can be a different area of research, whether trends and pattern detection can be automated.

- **Hybrid approach:** E.g. Template-based approach can be applied to add predefined texts to the final reports. This can be done in the early stage of the prototype or in the later part. This can be beneficial when the report has predefined report headings and footer text.

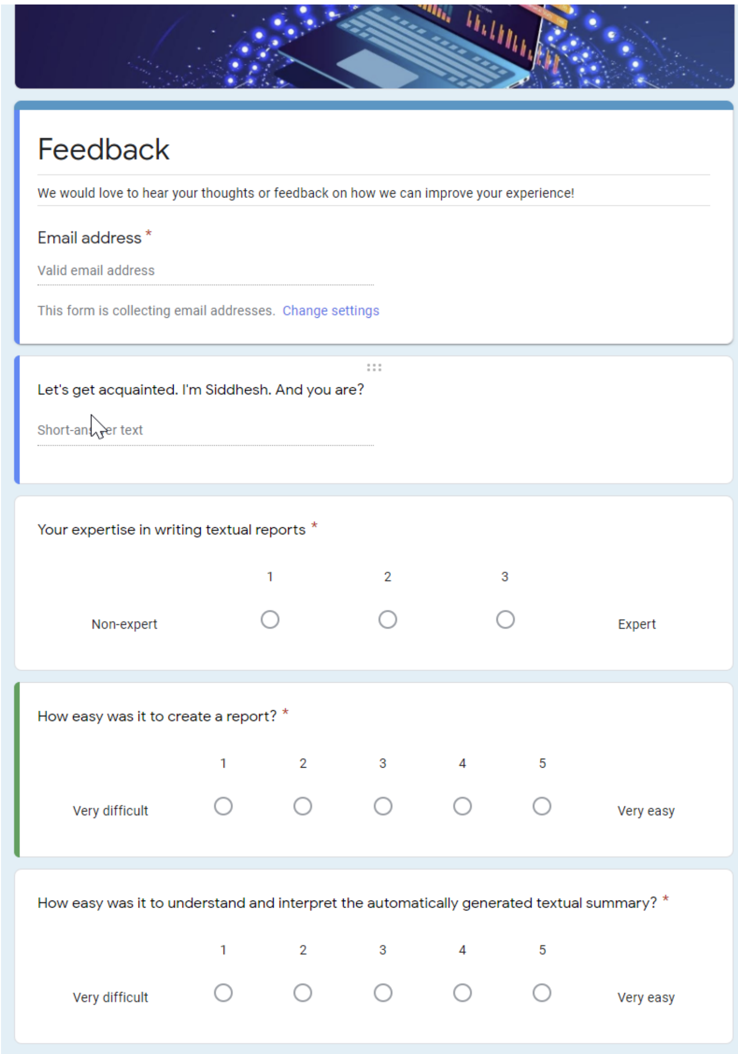
## 8.3. Conclusion

In this thesis, I proposed a rule-based NLG prototype for time-series KPI data, which overcomes the issues with manual report writing. The prototype developed is cost-effective and easy to implement, and serves as a good alternative to costly commercial rule-based NLG tools. The rule-based approach works well in our case study, with KPI data with less number of data points. But, this approach might not be that effective for complex data sets and complex domains. As discussed in previous section, it will be interesting to see if hybrid NLG system can be developed which can handle more complex data for varying domains, and is also cost-effective and easy to implement at the same time.

# Appendix

# A. Online survey questionnaire

Following as the list of questions which were asked in the online survey conducted for prototype evaluation.


Feedback

We would love to hear your thoughts or feedback on how we can improve your experience!

Email address *

Valid email address

This form is collecting email addresses. Change settings

Let's get acquainted. I'm Siddhesh. And you are?

Short-answer text

Your expertise in writing textual reports *

| | 1 | 2 | 3 | |
|---|---|---|---|---|
| Non-expert | ○ | ○ | ○ | Expert |

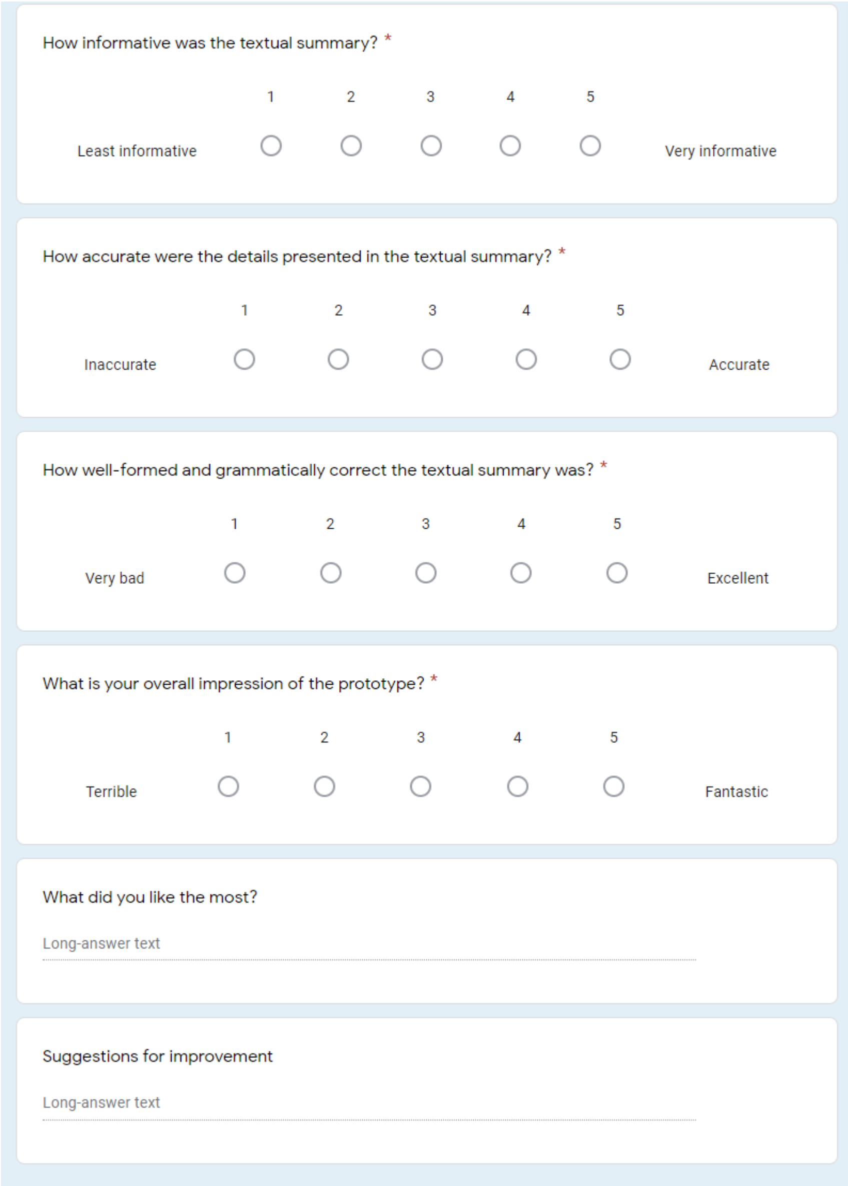
How easy was it to create a report? *

| | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| Very difficult | ○ | ○ | ○ | ○ | ○ | Very easy |

How easy was it to understand and interpret the automatically generated textual summary? *

| | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| Very difficult | ○ | ○ | ○ | ○ | ○ | Very easy |

Figure A.1.: RepGen evaluation: Survey questionnaire - part 1



How informative was the textual summary? *

| | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| Least informative | ○ | ○ | ○ | ○ | ○ | Very informative |

How accurate were the details presented in the textual summary? *

| | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| Inaccurate | ○ | ○ | ○ | ○ | ○ | Accurate |

How well-formed and grammatically correct the textual summary was? *

| | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| Very bad | ○ | ○ | ○ | ○ | ○ | Excellent |

What is your overall impression of the prototype? *

| | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| Terrible | ○ | ○ | ○ | ○ | ○ | Fantastic |

What did you like the most?

Long-answer text

Suggestions for improvement

Long-answer text

Figure A.2.: RepGen evaluation: Survey questionnaire - part 2

# B. Task-based interviews

Following are some of the textual executive summaries written by the interviewees.

- Interviewee name : Ashwin Prabhu
  Occupation: Software developer
  Expertise in reporting: No

  1. Write a textual summary for Electricity consumption in the year 2017.
     **The total electricity consumption for the year 2017 is 1236.63kWh with a monthly average consumption of 103.05 kWh. The max electricity consumption was in the month of Jan with consumption unit of 120.27 kWh and the lowest being the 89.75 kWh for the month of April.**
  2. Compare Electricity consumption in 2017 and 2018. Write a textual summary for the same.
     **When we compare the pattern of consumption its almost the same for both the years. The maximum consumption if electricity is in the month of Jan and minimum consumption is done in the month of April. The consumption for both the year is almost the same.**
  3. Write a textual summary on unique/contrasting data observations for the entire time period (1st January 2017 to 1st August 2020).
     **The maximum consumption if electricity is in the month of Jan and minimum consumption is done in the month of April. For 2020 there was a sudden dip in the power consumption for the month of June and July and then a raise.**

- Interviewee name : Gerardo Hernandez
  Occupation: Customer sales manager
  Expertise in reporting: Expert

  1. Write a textual summary for Electricity consumption in the year 2017.
     **For the year 2017 the total electricity consumption was 1,236.63 kWh. The average electricity consumption for that year was 103.05 kWh. The peak of electricity consumption, was in January with 120.27 kWh, and the lowest consumption was in April with 89.75 kWh.**
  2. Compare Electricity consumption in 2017 and 2018. Write a textual summary for the same.
     **For 2018 the behavior of electricity consumption is quite similar with small variances. January was also the peak and April also the lowest consumption**
  3. Write a textual summary on unique/contrasting data observations for the entire time period (1st January 2017 to 1st August 2020).
     **There is a particular trend in the period from January to April for the year 2017 where consumption slope was higher compared to 2018.**

- Interviewee name : Kasturi More
  Occupation: Consultant
  Expertise in reporting: Basic

1. Write a textual summary for Electricity consumption in the year 2017.
   **The total electricity consumption for the year 2017 is 1236.63 kWh. Every month the electricity consumption varies however, it is the decrease or increase in the usage is not consistent. However, as seen in the graph, the average usage is 103.05 kWh over the year. The highest consumption was seen in the month of January i.e. 120.27 kWh and the lowest in the month of April i.e. 89.75 kWh.**
2. Compare Electricity consumption in 2017 and 2018. Write a textual summary for the same.

| Factors | Electricity consumption in 2017 (kWh) | Electricity consumption in 2018 (kWh) |
|---|---|---|
| Total consumption | 1236.63 | 1233.17 |
| Average consumption/year | 103.05 | 102.76 (kWh) |
| Highest consumption | 120.27 | 117.08 |
| Lowest consumption | 89.75 | 89.33 |

Table B.1.: Brief summary on the electricity consumption for the years 2017 and 2018

3. Write a textual summary on unique/contrasting data observations for the entire time period (1st January 2017 to 1st August 2020).
   **The average consumption for the years 2017, 2018, and 2019 seems quite consistent. However, there is a drastic difference in the consumption of electricity for the year 2020 i.e. the average consumption was 441.05 kWh. It can be observed that from the month of March to May, the consumption of electricity has dropped and then remained constant for the months May to July. The reduction in the consumption is due to the pandemic attack by the COVID-19 across the world. The lockdown of the companies or allowing work from home opportunities have highly influenced this number.**

Following are the some of the observations and participants feedback from task-based interviews:

- Average time taken to write executive summaries for all the tasks was 15 minutes.
- Problem statement and tasks were easy to understand.
- Time-series KPI data set and time-series line chart was easy to interpret.
- Writing a textual executive summary was more challenging than interpreting the data.
- Deciding what to include and exclude in the textual summary and organizing that content of the textual summary was challenging.
- Automation can improve the overall process of writing textual executive summaries.

# Glossary

**Data experts** The person(s) who are able to understand, process and analyze the data. And use the hidden data insights to drive business decisions. Data analysts, data scientists, business analysts, consultants often portray the role of a data expert.. iv

**RepGen** It is an automated reporting assistant prototype developed to assist data experts in writing textual executive summaries for data reports. It is developed as a single page application in dash and plotly. iv

# Acronyms

**BI** Business Intelligence. 3

**IBC** Intelligent Business Cloud. 3, 7

**KPI** Key Performance Indicator. iv, 3, 6

**NLG** Natural Language Generation. 6

**NLU** Natural Language Understanding. 12

# Bibliography


[1] J. Bulao. *How Much Data Is Created Every Day in 2020?* URL: `https://techjury.net/blog/how-much-data-is-created-every-day/#gref`. (accessed: 01.09.2020).

[2] *About Celonis*. URL: `https://www.celonis.com/company/`.

[3] E. Brynjolfsson and Kim. "Strength in Numbers: How Does Data-Driven Decision making Affect Firm Performance?" In: (Apr. 2011). URL: `http://dx.doi.org/10.2139/ssrn.1819486`.

[4] E. Rahm and H. H. Do. "Data cleaning: Problems and current approaches". In: (2000).

[5] *Celonis Snap*. URL: `https://www.celonis.com/solutions/celonis-snap`.

[6] E. Reiter and R. Dale. "Building Applied Natural Language Generation Systems". In: (May 1997), p. 1.

[7] D. McDonald. "Issues in the Choice of a Source for Natural Language Generation". In: (1993). URL: `https://www.aclweb.org/anthology/J93-1009.pdf`.

[8] A. Mishra, A. Laha, K. Sankaranarayanan, P. Jain, and S. Krishnan. "Storytelling from Structured Data and Knowledge Graphs An NLG Perspective". In: (2019).

[9] A. Gatt and E. Krahmer. "Survey of the State of the Art in Natural Language Generation: Core tasks, applications and evaluation". In: (Jan. 2018).

[10] R. Dale. "The return of the chatbots". In: (Aug. 2016). URL: `https://www.researchgate.net/publication/308085341_The_return_of_the_chatbots`.

[11] B. Katz, J. J. Lin, D. Loreto, W. Hildebrandt, M. W. Bilotti, S. Felshin, A. Fernandes, G. Marton, and F. Mora. "Integrating web-based and corpus-based techniques for question answering". In: *n Proceedings of the Text Retrieval Conference (TREC)* (2003), pp. 426–435.

[12] E. Goldberg, N. Driedger, and R. Kittredge. "Using Natural Language Processing to Produce Weather Forecasts". In: (1994), pp. 45–53.

[13] D. Chen and R. Mooney. "Learning to sportscast: a test of grounded language acquisition". In: (July 2008), pp. 128–135.

[14] A. Law, Y. Freer, J. Hunter, R. Logie, N. Mcintos, and J. Quinn. "A Comparison of Graphical and Textual Presentations of Time Series Data to Support Medical Decision Making in the Neonatal Intensive Care Unit". In: (June 2005).

[15] E. Reiter, S. Sripada, J. Hunter, J. Yu, and I. Davy. "Choosing words in computer-generated weather forecasts". In: (Aug. 2005), pp. 137–169.

[16] K. Kukich. "Design of a knowledge-based report generator". In: (1983).

[17] S. Busemann and H. Horacek. "Generating Air Quality Reports From Environmental Data". In: (Oct. 1997). URL: https://www.researchgate.net/publication/2701957_Generating_Air_Quality_Reports_From_Environmental_Data.

[18] J. Coch. "Interactivegeneration and knowledge administartion in MultiMeo". In: (Oct. 1998). URL: https://www.aclweb.org/anthology/W98-1435.pdf.

[19] E. Goldberg, N. Driedger, and R. Kittredge. "Using natural-language processing to produce weather forecasts". In: *IEEE Expert* (Apr. 1994). DOI: 10.1109/64.294135.

[20] B. Sigurd, C. Willners, C. Johansson, and M. Eeg-Olofsson. "Deep comprehension, generation and translation of weather forecast (WEATHRA)". In: (Aug. 1992).

[21] S. Boyd. "TREND: A System for Generating Intelligent Descriptions of Time-Series Data". In: *IEEE International Conference on Intelligent Processing Systems* (1998). DOI: 10.1.1.57.3705.

[22] E. Manishina. "Data-driven natural language generation using statistical machine translation and discriminative learning". In: (July 2017).

[23] A. Leuski and D. Traum. "A tool for building question-answering characters". In: *Proceedings of the Language Resources and Evaluation Conference (LREC)* (2010).

[24] C. Shannon. "A Mathematical Theory of Communication". In: (July 1948). DOI: https://doi.org/10.1002/j.1538-7305.1948.tb01338.x.

[25] L. Sha, L. Mou, T. Liu, P. Poupart, S. Li, B. Chang, and Z. Sui. "Order-Planning Neural Text Generation From Structured Data". In: (2010).

[26] D. Pawade, A. Sakhapara, M. Jain, N. Jain, and K. Gada. "Story Scrambler - Automatic Text Generation Using Word Level RNN-LSTM". In: (June 2018). DOI: 10.5815/ijitcs.2018.06.05.

[27] W. Yonghui, S. Mike, C. Zhifeng, N. Mohammad, M. Wolfgang, K. Maxim, C. Yuan, and G. Qin. "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation". In: (Sept. 2016). URL: arXiv:1609.08144.

[28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, and I. Polosukhin. "Attention Is All You Need". In: (Dec. 2017). URL: arXiv:1706.03762v5.

[29] J.-S. Lee and J. Hsiang. "Patent Claim Generation by Fine-Tuning OpenAI GPT-2". In: (July 2019). URL: arXiv:1907.02052.

[30] E. Reiter. *SimpleNLG*. URL: https://github.com/simplenlg/simplenlg/wiki.

[31] A. Gatt and E. Reiter. "SimpleNLG: A realisation engine for practical applications". In: (Mar. 2009), pp. 90–93. URL: https://www.aclweb.org/anthology/W09-0613.pdf.

[32] L. Stoeckle. *RosaeNLG Documentation*. URL: https://rosaenlg.org/rosaenlg/1.17.0/index.html.

[33] I. A. Dimitrios G. George K. and G. L. "An Open-Source Natural Language Generator for OWL Ontologies and its Use in Proteg´ e and Second Life". In: (Apr. 2009), pp. 17–20. URL: https://www.aclweb.org/anthology/E09-2005/.

[34] G. Dimitrios, A. Ion, and L. Gerasimos. "Generating Natural Language Descriptions from OWL Ontologies: the NaturalOWL System". In: (Nov. 2013), pp. 671–715. URL: https://arxiv.org/abs/1405.6164.

[35] *Requirements Engineering Specialist Group*. URL: https://en.wikipedia.org/wiki/Requirements_Engineering_Specialist_Group#cite_note-RESG-1.

[36] D. Pandey. "Requirement Engineering Research". In: (May 2013).

[37] R. Ramos and A. Kurts. "Requirements Engineering A good practice guide". In: (1997).

[38] S. G. Gunda. "Requirements Engineering: Elicitation Techniques". In: (2008).

[39] B. Nuseibeh and S. Easterbrook. "Requirements Engineering: A Roadmap". In: (2000). URL: http://mcs.open.ac.uk/ban25/papers/sotar.re.pdf.

[40] *Celonis Snap*. URL: https://academic-siddhesh-kandarkar-tum-de.eu-2.celonis.cloud/process-mining/ui.

[41] K. Stichwörter. *Problem Statements and Design Goals*. URL: https://wiki.fluidproject.org/display/fluid/Problem+Statements+and+Design+Goals.

[42] C. Paris, D. Scott, N. Green, K. McCoy, and D. McDonald. "Desiderata for Evaluation of Natural Language Generation. In Shared Tasks and Comparative Evaluation on Natural Language Generation". In: (2007).

[43] A. Belz and E. Reiter. "Comparing automatic and human evaluation of nlg systems". In: (2006).

[44] A. Belz and E. Kow. "System Building Cost vs. Output Quality in Data-To-Text Generation". In: (Mar. 2009). URL: `https://www.aclweb.org/anthology/W09-0603`.