

# **REAL-TIME CARBON NEUTRALITY MANAGEMENT AND OPTIMIZATION USING NATURAL LANGUAGE PROCESSING**

Paskaran Sathees, Mannavarasan Mathanika, Sivarajah Vishakanan,  
Magenthirarajah Vithursan

(IT19052748, IT19005218, IT19001562, IT19033174)

BSc (Hons) in Information Technology

Department of Information Technology

Sri Lanka Institute of Information Technology  
Sri Lanka

September 2022

# **REAL-TIME CARBON NEUTRALITY MANAGEMENT AND OPTIMIZATION USING NATURAL LANGUAGE PROCESSING**

Paskaran Sathees, Mannavarasan Mathanika, Sivarajah Vishakanan,  
Magenthirarajah Vithursan

(IT19052748, IT19005218, IT19001562, IT19033174)

Dissertation Submitted in Partial Fulfillment of the Requirements for the BSc (Hons)  
in Information Technology


Department of Information Technology

Sri Lanka Institute of Information Technology  
Sri Lanka

September 2022

**Declaration**

We declare that this is my own work, and this proposal does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or institute of higher learning, and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text. Also, we hereby grant to Sri Lanka Institute of Information Technology the non-exclusive right to reproduce and distribute my dissertation in whole or part in print, electronic or other medium. We retain the right to use this content in whole or part in future works (such as article or books).

Name	Student ID	Signature
Sathees P	IT19052748	
Mathanika M	IT19005218	
Vishakanan S	IT19001562	
Vithursan M	IT19033174	

Signature of the supervisor:

Date:

## **Abstract**

Greenhouse gas (GHG) emissions from human activities contribute to climate change. Nowadays, governments worldwide enforce many regulations to control GHG emissions. As a result, organizations must monitor their GHG emissions and report them to their respective governments. Moreover, organizations should limit their emissions to the allowed cap (Carbon credit) to achieve carbon neutrality. For this purpose, many governments release emission factors for different emission sources, which organizations use to calculate emission values. However, this is a tedious and error-prone workload for organizations' business analysts, which includes keeping track of every emission task, finding relevant emission factors, and calculating emissions. Moreover, extensive analysis is needed to optimize emissions. Therefore, the proposal expects to implement a cross-platform mobile application to collect emission tasks from the employees in real-time in natural language input. These inputs will find the most relevant emission factors using information retrieval and natural language processing. A unit conversion process will check and convert the user input units to match the emission factor's units before calculating the emission. Emission data can be accessed in real-time using a business intelligence tool. The system will provide an optimal solution to minimize emissions using the emission constraints and send alerts regarding optimal solution violations.

**Keywords:** Carbon Footprint Management, Natural Language Processing, Named Entity Recognition, Word Embedding, Vector Space Model, Text Classification, Linear Programming.

## **Acknowledgment**

We would like to express our sincere gratitude to all those who helped us accomplish this dissertation, especially our supervisors, Ms. Anjalie Gamage and Ms. Sanjeevi Chandrasiri of the faculty of computing, Sri Lanka Institute of Information Technology. Moreover, we would like to thank the panel members Prof. Koliya Pulasinghe, Ms. Rubaa Panchendrarajan, and Mr. Vishan Jayasinghearachchi, faculty of computing, Sri Lanka Institute of Information Technology, for their valuable comments. In addition, we thank our external supervisor Dr. Daniel N Subramaniam, faculty of engineering, university of Jaffna, for the practical domain knowledge.

## **Table of Contents**

Declaration	iii
Abstract	iv
Acknowledgment	v
Table of Contents	vi
List of Figures	viii
List of Tables	ix
List of Abbreviations	x
List of Equations	xi
1. INTRODUCTION	1
1.1. Background and Literature Survey	1
1.2. Research Gap	4
2. RESEARCH PROBLEM	6
3. OBJECTIVES	10
3.1. Main Objectives	10
3.2. Specific Objectives	10
4. METHODOLOGY	12
4.1. Preliminaries	12
4.1.1. Emission calculation	12
4.1.2. Named entity recognition (NER)	12
4.1.3. Vector space model (VSM) and word embedding	14
4.1.4. Text classification and regex	16
4.1.5. Linear programming	18
4.2. Complete System Architectures	19
4.3. Data Collection	19
4.3.1. Emission activity parts extraction	20
4.3.2. Emission factor retrieval	20
4.3.3. Unit verification and conversion	21
4.3.4. Emission optimization	21
4.4. Technologies and Implementation	22
4.4.1. Emission activity parts extraction implementation	22

4.4.2.	Emission factor retrieval implementation	22
4.4.3.	Unit verification and conversion implementation	23
4.4.4.	Emission optimization implementation	25
4.4.5.	Application backend implementation	26
4.4.6.	Application frontend implementation	28
4.5.	Experimentations	28
4.5.1.	Emission activity parts extraction evaluation	19
4.5.2.	Emission factor retrieval evaluation	28
4.5.3.	Unit verification and conversion evaluation	20
4.5.4.	Emission optimization evaluation	29
4.6.	Commercialization	30
5.	RESULTS AND DISCUSSION	33
5.1.	Results	33
5.1.1.	Emission activity parts extraction results	33
5.1.2.	Emission factor retrieval results	34
5.1.3.	Unit verification and conversion results	27
5.1.4.	Emission optimization results	37
5.2.	Research Findings	38
5.2.1.	Emission activity parts extraction finding	27
5.2.2.	Emission factor retrieval finding	38
5.2.3.	Unit verification and conversion finding	39
5.2.4.	Emission optimization finding	40
5.3.	Discussion	40
5.4.	Summary of Each Student Contribution	41
6.	CONCLUSIONS	44
	REFERENCES	47
	APPENDICES	51
	Appendix A. UI Prototypes	51

## List of Figures

Figure 4.1: Visualization of a word vector (“glove-wiki-gigaword-300”) for terms related to traveling mediums. ....	15
Figure 4.2: Simplified complete system architecture .....	19
Figure 4.3: Complete system architecture .....	19
Figure 4.4: Application database Entity Relation (ER) diagram .....	26
Figure 4.5: Data warehouse physical diagram.....	27
Figure 4.6: Business model canvas .....	30
Figure 4.7: Pricing plan .....	31
Figure 4.8: Product landing page.....	31
Figure 4.9: Product promotion pamphlet.....	32
Figure 5.1: Optimization results visualization for user satisfaction using MAP.....	35
Figure 5.2: Average CPU time with term count.....	35
Figure 5.3: Average CPU time with document count .....	35
Figure 5.4: Average CPU time with word vector and delta values .....	36



## List of Tables

Table 1.1: Previous research and products comparison .....	4
Table 4.1: EF ranking module implementation technologies and usage .....	22
Table 4.2: Backend implementation technologies and usage .....	27
Table 4.3: Mobile frontend development technologies and usage .....	28
Table 5.1: Average query time for vast EF datasets .....	36
Table 5.2: Memory and storage usage by TF-IDF matrices .....	36

## List of Abbreviations

Abbreviation	Description
CRIS	Climate Registry Information System
DEFRA	Department for Environment, Food and Rural Affairs
EPA	Environmental Protection Agency
IPCC	Intergovernmental Panel on Climate Change
NGA	National Greenhouse Accounts
NDC	Nationally Determined Contribution
GHG	greenhouse gases
EF	Emission Factor
UOM'	Unit of Measure
CO2-eq	CO2-equivalent
GWP	Global Warming Potential
VSM	Vector Space Model
IR	Information Retrieval
TF-IDF	Term Frequency-Inverse Document Frequency
BI	Business Intelligence
AWS	Amazon Web Services
API	Application Programming Interface
UI	User Interface
MAP	Mean Average Precision
MB	MegaByte
CPU	Central Processing Unit

## List of Equations

### Equation

(1) Emission calculation

2

# **1. INTRODUCTION**

## **1.1. Background and Literature Survey**

Concerns about the impacts of greenhouse gas (GHG) emissions on climate change have been growing significantly over recent years [1]. Due to these concerns, many governments and supra-national bodies such as United Nations (UN) and European Union (EU) have formed policies and agreements to encourage reducing GHG emissions [2]. Significant of these policies were Kyoto Protocol in 1997 and Paris Agreement in 2016 [1], [2]. The Paris agreement proposed stricter measures which target to limit global warming increase below 2 degrees Celsius, preferably to 1.5 degrees Celsius [1]–[5]. The countries require their planned actions to contribute to the Paris agreement aim every five years; these are the Nationally Determined Contributions (NDCs) [1]. As one of the parties to the Paris Agreement, Sri Lanka expects to achieve carbon neutrality by 2060 and has taken many steps to achieve it, such as reducing emissions related to electricity production by using renewable energy [6]. It is also noteworthy that GHG emissions from industries are a significant contributor to global warming [7]. To accommodate NDCs, governments have implemented many national climate regulations methodologies such as carbon tax, carbon trade-off, carbon cap-and-trade, mandatory carbon reporting, and carbon emission disclosure (also known as Carbon Disclosure Project (CDP)) by the firms [8], [9]. These measures eventually led to the birth of the carbon trading paradigm.

In carbon trading, firms purchase or obtain a certain amount of carbon credits. Carbon credits allow firms to emit a certain amount of GHG emissions. However, it is beneficial for firms to stay within their available credit limit. Therefore, firms balance their total GHG emissions for a period with their available carbon credits [10]. This process is carbon accounting or carbon reporting. Firms achieve carbon neutrality by staying within their credit limits [11]. These carbon trading implementations provide significant benefits, encouraging consumers, companies, and managers to find sustainable alternatives even if they are expensive [12]. Moreover, it also promotes technological innovation and competitiveness in implementing sustainable alternatives [9].

Regardless of the carbon trading schemes used, firms must measure GHG emissions. GHG emission occurs when an emission activity (e.g., usage of 1kWh of electricity) is carried out in the firm [13]. The GHG emissions measurement units include Metric tons or kilograms. During an emission activity, various GHGs, such as carbon dioxide, methane, nitrous oxide, and hydrofluorocarbons, are released, and these GHGs have different global warming potentials [2], [14]. Global warming potential estimates how much each GHG contributes to global warming compared to carbon dioxide [14]. However, conducting carbon reporting for all these gases would complicate the carbon accounting process. Therefore, the organization usually uses a standard measure known as **CO<sub>2</sub> equivalent** (CO<sub>2</sub>e) to measure GHG emission for different emission sources [1], [13]. These CO<sub>2</sub>e values for different emission sources are estimated and published in different document formats by different environmental entities (also known as **emission factors**) [13]. Firms usually adopt one of these emission factors as their standard for carbon reporting depending on their reporting jurisdiction. Some of those standards are published by,

- Department for Environment, Food and Rural Affairs (DEFRA) [15] – UK
- Climate Registry Information System (CRIS) [16] – USA and Canada
- Environmental Protection Agency (EPA) [17] – USA
- National Greenhouse Accounts (NGA) [18] – Australia

GHG emission for an emission activity is simply a product of consumption with a specific emission factor (1) [13], [19]. For example, if we assume the emission factor for an average car is 0.1500 kgCO<sub>2</sub>e/km, we have traveled 4 km using this car. Calculated GHG emission = 4 km × 0.1500 kgCO<sub>2</sub>e/km = 0.6 kgCO<sub>2</sub>e.

$$GHG\ emission = consumption \times specific\ emission\ factor \quad (1)$$

Moreover, emission activities scope classification includes Scope 1, Scope 2, and Scope 3. Scope 1 includes emissions from direct sources owned by the firm, such as vehicles. Scope 2 includes emissions from indirectly purchased sources such as

electricity. Finally, scope 3 includes emissions from third-party indirect sources, such as waste management [4], [13], [14].

Current approaches for managing carbon emissions within firms are done either manually by a business analyst (BA) or using a commercial emission calculator. This study observed during this literature survey that there is an inadequacy in the research related to implementing corporate-level carbon emissions management. Additionally, some commercially available tools calculate and manage emissions [20]–[22]. However, current carbon reporting implementations have some gaps and issues [1], [5]. During the discussion with the industry expert, it has become evident that most of the current systems expect emission data to be collected and provided by the BA of the firm. However, there can be many data collection issues in this approach. Moreover, emission data collection usually happens at the end of the reporting period, and this causes a delay in reporting. Due to this delay, there might be chances of unanticipated scenarios like emitting more than the desired targets.

The proposed real-time carbon neutrality management system solves the current issues of the carbon reporting system. The proposed system will collect emission activity data from the firm's employees using a natural language input in real-time. The extraction component will be able to annotate and differentiate parts such as emission technology, date, consumption value, and consumption unit of the emission activities provided by the employees. Relevant emission factors for the emission technology (e.g., vehicle, generator, and diesel) are needed to proceed with emission calculation. However, for accurate emission calculation, emission factors for each of these activities should be selected carefully [13]. It is time-consuming to refer to these emission factor files to find relevant emission factors and requires significant domain knowledge about carbon reporting. During this process, the system will check units of these values before emission calculation. In addition, most organizations must achieve their emission goals while doing carbon reporting. Therefore, carbon footprint management systems need extraction with emission activity parts extraction, a more straightforward emission factor selection interface, unit checking and conversion, and an emission goal tracking feature.

## 1.2. Research Gap

The literature survey shows a necessity for more research in corporate carbon reporting implementation, and most of the available research concentrates on the effectiveness of various reporting schemes. In this Research A [23], the real-time carbon accounting method is only implemented for scope two emissions and does not focus on corporate-level emission management. This research does not feature an emission search system as well. However, there are many commercial systems implemented to solve the same issue, such as Product A [20], Product B [21], and Product C [22]. All these products require getting emission data from a single person or group of firm personnel using the user interface (UI) or file uploads in a specified file format.

Moreover, these systems do not provide real-time or timely emission status due to the delay in data collection. These products also do not feature an emission factor search system. All these commercial systems feature some sort of emission calculation.

*Table 1.1: Previous research and products comparison*

<b>Research or Product</b>	<b>Emission calculation</b>	<b>Data collection from employees</b>	<b>Emission factor search system</b>	<b>Ad-hoc emission factor searching</b>	<b>Emission factor ranking using term similarity and personalization</b>
<b>Research A</b>	✓	X	X	X	X
<b>Product A</b>	✓	X	X	X	X
<b>Product B</b>	✓	X	X	X	X
<b>Product C</b>	✓	X	X	X	X
<b>Proposed System (Carbonis)</b>	✓	✓	✓	✓	✓

From research and product comparison, the novelty of the proposed system (Carbonis) is justifiable as there were no previous works within this domain that feature emission data collection from employees, emission factor search system, and emission

optimization. In the domain of carbon reporting technologies, observation suggests that there was no previous work on emission factor search systems, proving the component's novelty separately. It is noteworthy that there can be many benefits of using an ad-hoc emission factor search system with personalization in any carbon reporting tool, such as,

- Timesaving

Searching can be faster than traversing through files or using dropdown menus. In this case, the system must find emission factors within thousands of emission factors in different documents. Therefore, using this search approach is expected to increase carbon reporting performance.

- Less fatigue

Scrolling through files can cause fatigue to employees, which could lead to a bad user experience in emission reporting. As searching can be less fatigue, employees will be more willing to report their emissions.

- Tolerant to various representations of the search queries

Employees can give search queries for their information needs in different representations. The Ad-hoc search approach can be considered tolerant of this issue.

The emission search system is relevant for the proposed system (Carbonis) because emission activity and consumption units can be in different representations. There is no other suitable retrieval approach that is practical in this scenario. The proposed system consists of many beneficial features compared to previous research and currently available products in the market. It is unique in the way of data collection from the employees using natural language input. It includes an emission factor searching component that is tolerant of various representations of terms and can rank results using term similarity and personalization.



## 2. RESEARCH PROBLEM

In realizing the Paris agreement to lessen the consequences of climate change, legislation for a drastic reduction of greenhouse gas (GHG) emissions was imposed by governments worldwide [24]. As a part of this measure, organizations must disclose their GHG emissions publicly [9]. Some governments implement strategies such as carbon credit (cap on total GHG emissions that businesses are allowed to emit. E.g., one carbon credit = 1 metric ton of GHG emission), carbon offsetting (businesses pay to have the emission reduced somewhere else. E.g., Solar power projects), and carbon in setting (investing in emission reduction within the business supply chain. E.g., In premise solar power supply). Businesses go "carbon neutral" by keeping the total emissions within the cap, and businesses go "carbon neutral." This reporting task is the organization's responsibility, usually handled by a business analyst (BA). For this, staff should record all emission tasks (e.g., use of 5l of gasoline for an electrical generator) and calculate emission values by comparing the relevant emission factor (e.g., 1l of gasoline will emit 0.25 kg of CO<sub>2</sub> equivalent) [13]. Governments will release the data sets of these emissions factors each year (e.g., DEFRA standard by the UK government) [25]. This task is overwhelming and requires a thorough understanding of this process. Moreover, there is no way of knowing the current emission compared to their target (emission cap). Therefore, there is a need for real-time monitoring and optimization of the organizations' GHG emissions [23].

Most countries' governments that are part of the Paris agreement [1] release quite an extensive emission factor document for various activities annually [25]. Final emission values calculation will happen using these factors for the activities performed over the reporting period [25]. However, it is not easy to find relevant emission factors as they are mentioned in technical terms and require a deeper understanding. Even though there are several commercially available emission calculators available, those also require users to have in-depth knowledge about the emission factors and want the users to select relevant emission factors [9]. As emission factors must be found for each activity, reducing the effort and time involved here would result in increased productivity for carbon reporting.

It is usually the responsibility of a single business analyst (BA) or a small BA team to maintain the emission activities of their organization, calculate emissions and produce reports. Collecting data on emission activities from various sources is a tedious task for a business analyst and might be erroneous on some occasions [2]. Since this is a time-consuming task and only performed a few times a month, there will be no up-to-date real-time status of the organization's emissions. This observation is also accurate for most of the current online emission calculators since they require users to take responsibility for data collection.

In the emission details collection using natural language inputs, users are not limited to entering specific units, and emission parts extraction would not be aware of the different units it only knows to classify as a unit. Emission factor search will also result in most matching and not limited to units, and users can select emission factors with different units. Therefore, before calculating emission values, user-given values' units should be verified with selected emission factors units and must be converted before calculating emission [26].

Even though reducing carbon emissions is a crucial task every organization must carry out, there are occasions when it is still not possible to consider alternative sustainable options. Most of these emission sources are related directly to the business process, and there will be no alternative option that will not affect the business's effectiveness. However, there may be other emission sources with a possible reduction in an organization to achieve their desired emission goal. Therefore, there should be a way to find an optimal solution for these constraints on emission sources.

An innovative solution proposed for the above scenario would be to implement a real-time platform that can provide insights into the most up-to-date emission statistics of the organization. Emission activity data will be directly gathered from the employees using a natural language input. For each emission activity input, the system should find relevant emission factors with the help of an information retrieval process. Before proceeding, it will clarify any misinformation with the employee. Furthermore, the

system will check units of inputs by using text classification with natural language processing, and values will be converted before emission calculation to avoid miscalculations. The system will store calculated emission values for analysis purposes, and business analysts can access this real-time data using any business intelligence tool. For the constraints provided on the emission sources and emission cap, the system should generate an optimal solution using optimization models (algorithms) and will send alerts if there is any breach of the optimal solution.

Using the emission activity parts identified by the emission activity parts extractor, a factor-searching module will provide relevant factors back to the extractor for confirmation. Since the emission factors are in a technical language, a natural language-based information retrieval system will be used to find relevant emission factors conveniently. It will find the relevancy of a factor to the query based on closeness, term frequency, and personalization weightage. This approach should be scalable to many different emission factor standards.

The proposal will implement a natural language-based emission activity parts extractor for faster input of emission tasks. Employees can provide emission records using natural languages. Necessary parts will be obtained from the audio file using natural language processing approaches. In the event of ambiguity or missing parts, the emission activity parts extractor will request the user to clarify. Moreover, it will get confirmation on emission factors found as relevant.

Classes of the unit provided by the user and the unit in the selected emission factor will be classified using a text classification approach using natural language processing. Using the identified classed similarity of the units will be verified if the classes are the same. If those classes are different, the conversion factor for unit change will be derived using a conversion matrix for different unit classes and will convert values using the conversion factor before calculating emission.

The business analyst will provide details on usage constraints on several emission sources. Optimization models will provide minimum optimal solutions for these

constraints and emission caps. Furthermore, business analysts can customize this optimization according to their requirements changes. Once an optimal solution is confirmed, the system will set the maximum emission value provided by the solution for various emission sources as the max threshold and will send alerts for the violations.

### **3. OBJECTIVES**

#### **3.1. Main Objectives**

The main objective of this work is to implement a cross-platform mobile application for organizations to manage and optimize their GHG emissions. This mobile application will be able to collect emission activity data from the employees of the organization. It should identify the emission activity parts needed for the emission calculation from these data. Moreover, the application should recommend emission factors relevant to the given emission activity data. Using these outputs application should be able to calculate emissions. During emission calculation system will check and convert units provided from emission activity data and emission factor units. Finally, it will be able to provide optimal solutions for the consumption constraints imposed on the emission sources and emission cap of the organization.

#### **3.2. Specific Objectives**

To achieve the main objective following four specific objectives needs to be achieved,

1. Gather employee emission activity details from employees using natural language inputs.

From the text input provided by the employee using the mobile application, this component should extract emission activity parts such as emission technology, consumption, consumption unit, date, and optionally emission source.

2. Search emission factors and provide ranked results based on closeness or similarity, term frequency, and personalization weightage.

This component should find relevant emission factors from emission factor datasets by considering emission activity data as the query and rank emission factors based on the term similarity, term frequency, and personalization weightage.

3. Verify and convert values for units provided by the employees to match the units of the selected emission factor.

This component should be able to verify whether the consumption unit provided by the employee and the emission factor unit match. If the units do

not match, it should convert the consumption value to the unit of emission factor.

4. Identify the optimum solution for the given emission source constraints and alert about any violations of the optimal solution.

This component should find the minimal optimal solution for the minimum consumption emission source constraints imposed on the emission sources and the emission cap set by the organization. Then, by storing these optimal solution values as the thresholds, it will verify the occurrence of any violations and will send alerts to relevant personnel on those violations.

## **4. METHODOLOGY**

Methodology discusses preliminaries for technologies used, complete system architectures, data collection procedure, technologies and implementation details, experimentations, and commercialization aspects.

### **4.1. Preliminaries**

Preliminaries give background information on emission calculation (used in component 2: emission factor ranking), named entity recognition (used in component 1: emission activity parts extraction), vector space model, and word embedding (used in component 2: emission factor ranking), text classification and regex (used in component 3: unit verification and conversion), and linear programming (used in component 4: emission optimization).

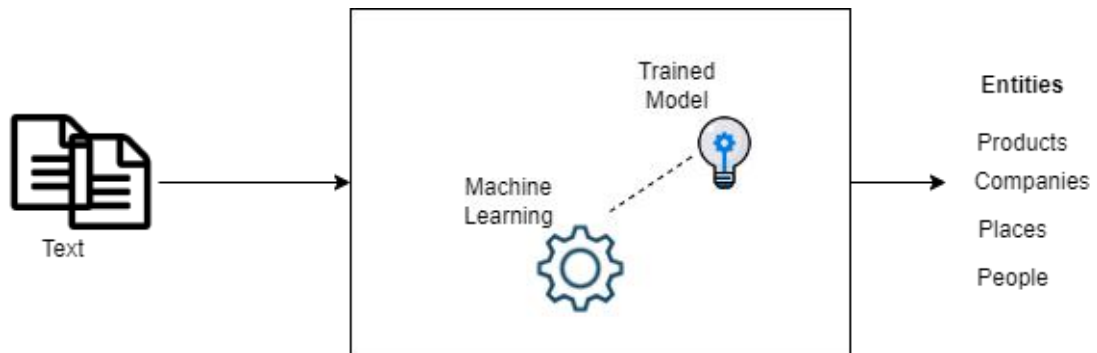
#### **4.1.1. Emission calculation**

GHG emission for an emission activity is simply a product of consumption and emission factor for emission technology (1) [13], [19]. However, there need to be considerations such as identifying the most suitable emission factor for that emission technology and ensuring the units of both consumption and emission factors are a match to ensure the accuracy of the calculation. For example, if we assume the emission factor for an average car is 0.1500 kgCO<sub>2</sub>e/km, we have traveled 4 km using this car. Calculated GHG emission = 4 km × 0.1500 kgCO<sub>2</sub>e/km = 0.6 kgCO<sub>2</sub>e. In a real-world scenario, the user must manually search the appropriate dataset to find the correct emission factor and convert units manually to ensure the calculation validity.

#### **4.1.2. Named entity recognition (NER)**

The term "Named Entity (NE)," which is widely used in Information Extraction (IE), Question Answering (QA), and other Natural Language Processing (NLP) applications, originated in the Message Understanding Conferences (MUC), which influenced IE research in the United States in the 1990s [Grishman and Sundheim 1996] (to be precise, it was first used in MUC-6 in 1995). MUC's concentration at the time was on IE tasks, which involved extracting structured information about company

activities and defense-related activities from unstructured text, such as newspaper articles.



*Figure 4-1: Action of Custom Named Entity Recognition*

Every day, the amount of text generated in various fields such as health care, news articles, scientific publications, and social media skyrockets. The International Data Corporation (IDC) has forecast that by 2020, the volume of data will have increased 50-fold to 40 billion gigabytes. Textual data is relatively frequent in most disciplines, but due to its unstructured nature, automated comprehension is difficult, which has led to the development of numerous text mining (TM) algorithms in the recent decade.

When it comes to custom named entity recognition in order to extract domain-specific entities from unstructured text, such as contracts or financial documents, users of custom NER can create their own AI models. Before making a model usable, developers can iteratively label data, train, assess, and enhance model performance by developing a Custom NER project. Performance of the model is heavily influenced by the quality of the labelled data. We require a source data of the entities with words when it comes to custom named entity recognition. only after that can we use our own words or entities to train the NER model, which will allow us to forecast the appropriate entity.

We are extracting the following emission parts: emission technology, emission date, emission source, emission value, and emission unit. NER tools are widely used by industries and are widely available for free. Hugging face, spaCy, Stanford NER, and Natural Language Toolkit (NLTK) The evaluation's goal was to determine whether the tool could identify names' borders and their proper types. Based on exact boundary



and type matching, we graded NER systems. The objective of named entity recognition is to assign a specific class to each token (word) in a phrase. A person, a location, an organization, etc. can all be identified by the most popular NER systems that are freely accessible online as per the above mentioned reading it was crystal clear that the custom named entity recognition in real time emission calculating concept will be of many advantages. Although there are researchers prevailing in the area of the emission calculation concept but there is no research prevalent regarding the real time emission calculating using real time employee's emission data.

#### **4.1.3. Vector space model (VSM) and word embedding**

IR tries to find documents that fulfill user needs from a collection of documents (called text corpus) by using relevancy calculated between user queries and documents [27], [28]. However, all IR models limit how users express their needs in queries that impact IR results. Moreover, an IR system's effectiveness depends on document storage and retrieval effectiveness. Several IR models with high retrieval effectiveness are available, and the popular choices include VSM and probabilistic models [27], [28]. VSM utilizes term frequency and inverse document frequencies compared to probabilistic models such as BM25, which could be helpful in the application of emission factor search functionality. However, VSM still needs the exact terms specified in the query for the search, which will require the user to know the emission factor labels.

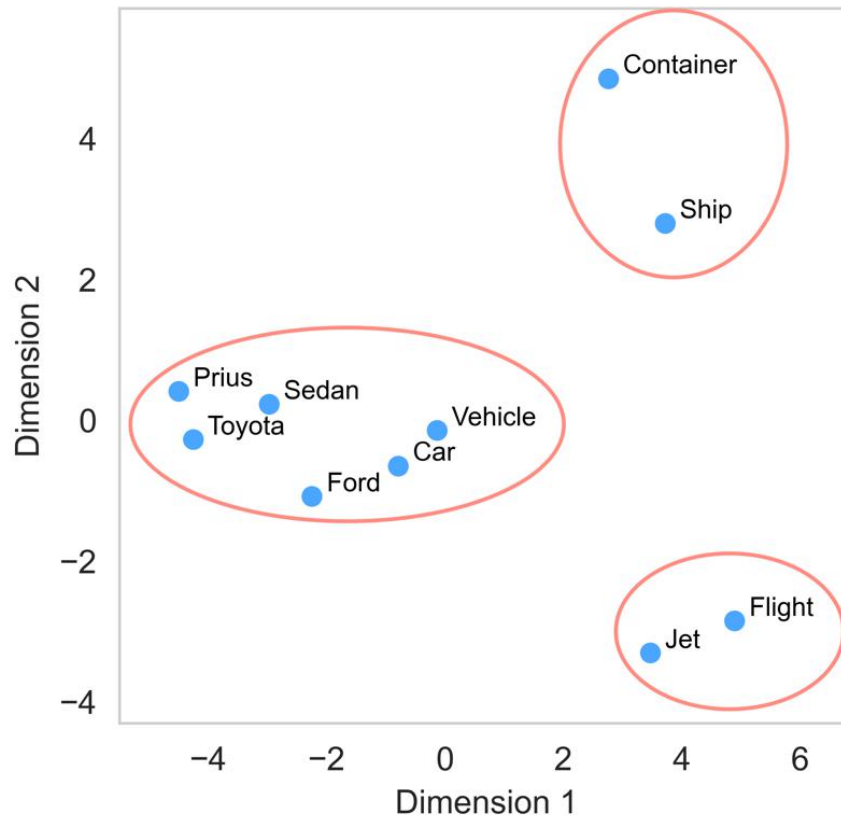


Figure 4-2: Visualization of a word vector (“glove-wiki-gigaword-300”) for terms related to traveling mediums.

A semantic representation is needed to decouple emission factor retrieval from emission factor labeling. The thesis implementation combines word embedding with VSM to provide a semantic understanding. As shown in Figure 4-2, word embedding assumes that co-occurring words have similar meanings [29], [30], and it captures semantic meanings of words within real number vectors [30]–[32]. Word embeddings tend to perform better than other count models [33]. It is a general practice to train deep learning models on a large text that stores knowledge on word vectors [29], [33]. These word vectors are byproducts of deep learning operations. Word embedding selection involves careful consideration of the embedding’s corpus domain, corpus size, training algorithm, and output vector dimension.

#### **4.1.4. Text classification and regex**

Text may be a tremendously rich source of information due to its unstructured nature, yet getting insights from it can be difficult and time-consuming. But as machine learning and natural language processing—both of which fall under the umbrella term of artificial intelligence—progress, sorting text data is becoming easier. It works by quickly and effectively independently analyzing and categorizing text, allowing businesses to automate procedures and uncover information that enhances decision-making.

Open-ended text is given a category by use of text categorization, a machine learning technique. Text files from the web, academic papers, and publications can all be organized, arranged, and categorized using text classifiers. Text categorization, one of the central issues in NLP, has several applications, including sentiment analysis, topic labeling, spam detection, and intent detection.

It is possible to manually or automatically categorize text. A human annotator is needed for manual text classification in order to assess the text's content and designate the proper category. Despite the fact that this process can yield positive outcomes, it is expensive and time-consuming.

Automatic text categorization classifies text more rapidly, effectively, and precisely using machine learning, natural language processing (NLP), and other AI-guided techniques. In this guide, we'll mostly focus on automated text classification. Although there are numerous techniques for automatically categorizing text, they always belong to one of three groups:

- systems based on rules
- Machine learning-based systems
- Hybrid systems

Out of those groups, I picked machine learning-based systems.

Instead of manually setting criteria, machine learning text categorization learns to create categories based on prior observations. By using pre-labeled examples as training data, machine learning algorithms may comprehend the numerous associations between text fragments and that a given output (tags) is predicted for a specific input (text). A "tag" is a predetermined classification or grouping that each provided text may fall into.

The first step in training an NLP classifier using machine learning is feature extraction, which entails converting each text into a numerical representation in the form of a vector. One of the most popular techniques is the "bag of words," in which a vector represents the frequency of a word within a specified lexicon.

The machine learning process is then used to generate a classification model from the training data, which consists of pairs of feature sets (vectors for each word) and tags..

If the machine learning model has enough training data to use in its training, it can begin making accurate predictions based on the image. The same feature extractor transforms unseen text into feature sets that can be fed into the classification model to generate predictions on tags.

Machine learning is often substantially more accurate than rule systems developed by humans for challenging NLP classification tasks. Machine learning classifiers are also easier to maintain, and you can always tag new examples to learn new techniques.

Deep learning algorithms have been shown to be especially good at identifying text, delivering cutting-edge results on a range of standard academic benchmark tests.

- Regex

The term "regex" or "regexp" refers to regular expressions, which are used to match text strings to certain letters, words, or character sequences. It indicates that any string pattern can be matched and extracted from text using regular expressions. I used the words "match" and "extract," and they both have quite distinct meanings. In certain situations, we might want to match a specific pattern but only extract a subset of it.

When looking through lengthy texts, emails, and documents, regex is quite helpful. Regex is referred to be a "programming language for the matching of strings." It's

crucial to understand regular expressions' practical usage before delving into their Python implementation. We can also transform mathematical operations using the Regex technique.

#### **4.1.5. Linear programming**

The Linear Programming Problems, sometimes known as LPP for short, are problems that involve determining the best possible value for a linear function that is provided. Either the highest possible value or the lowest possible value could be considered the best value. In this context, the linear function that has been supplied is regarded as an objective function. The objective function may include a number of variables that are governed by the conditions, and it must be able to meet a set of linear inequalities that are known as linear constraints. The linear programming issues can be used to achieve the optimal solution for the following scenarios, such as manufacturing problems, diet problems, transportation problems, allocation problems, and so on. The linear programming problems can also be used to solve other types of problems.

A linear programming approach was taken in the development of the system to neutralize carbon. The application that was produced via the use of optimization algorithms was then put through its paces by being tested with carbon emission data from a particular company in order to determine whether or not it was functioning appropriately or whether it had any limits. The optimization algorithm needed to reduce objective function as much as possible in order to meet the emission target. The computation of emissions was performed by considering the consumption of the emission activity and a particular emission factor. The amount of emissions can be calculated by taking the consumption and multiplying it by a specific emission factor. When measuring carbon emissions, kilograms of carbon dioxide are typically used as the benchmark.

## 4.2. Complete System Architectures

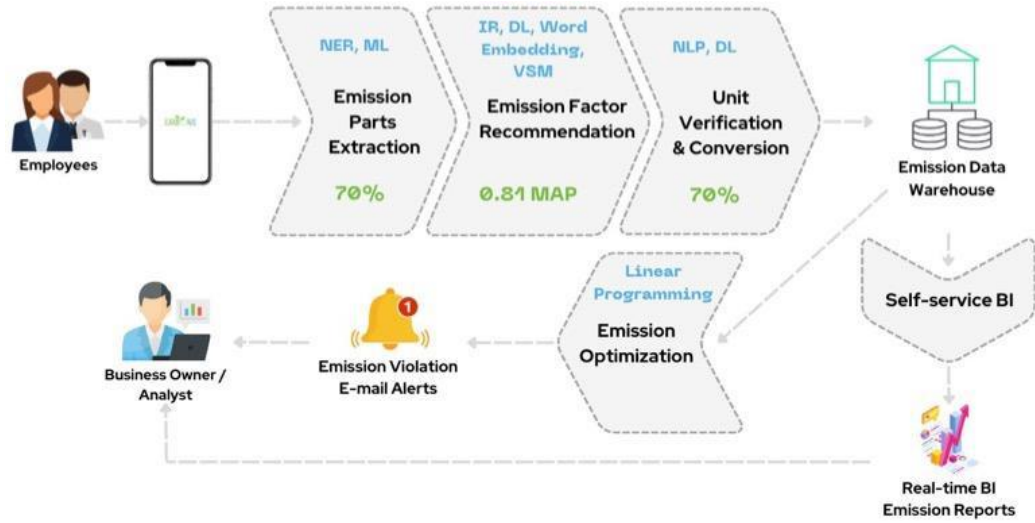


Figure 4-3: Simplified complete system architecture

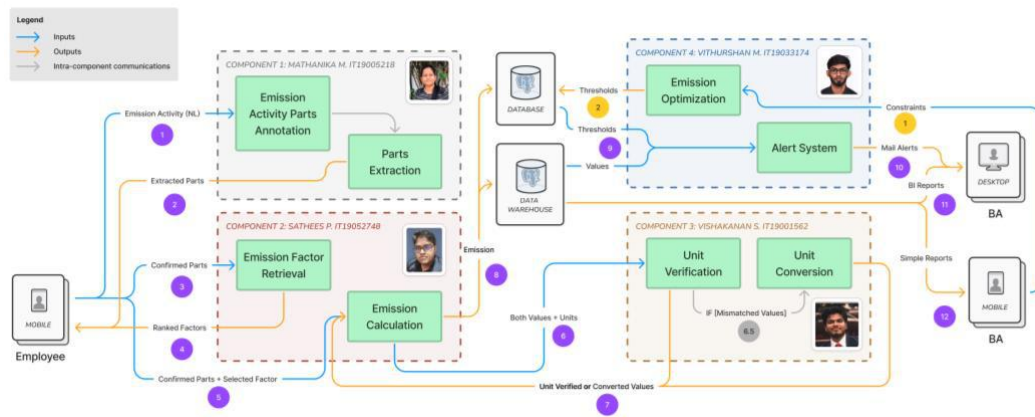


Figure 4-4: Complete system architecture

As shown in Figure 4-3 and Figure 4-4, there are four components: emission activity parts extraction, emission factor retrieval, unit verification and conversion, and emission optimization. Out of these four, emission activity parts extraction, emission factor retrieval, and unit verification and conversion will occur in the emission-adding process by all employees. However, only business analysts will access the emission optimization part.

## 4.3. Data Collection

Data collection discusses how the necessary datasets were gathered or generated for each of the four components' implementation.

#### **4.3.1. Emission activity parts extraction**

We require high-quality data to train the NER model before we can develop it. We gathered information for our emission extraction model through surveys and some common documents. We prepared surveys and distributed them to various client groups, including construction engineers, IT professionals, government officials, and municipality workers.

Additionally, we obtained data from certain emission standard publications to train the NER model. those are DEFRA, CRIS, EPA, NGA, IPCC. We have gathered over 500 different types of emission technologies and emission units from these emission standard publications.

There is not enough information gathered from surveys and papers to train the emission extraction model. In order to make the data obtained useable, some pre-processing is required. Each model requires a distinct type of data source in order to be trained.

To train our model, extraction models require annotated data formats. On the internet, there are several free programs that may be used for annotation. The emission factors are included in the annotated data source along with the entity.

#### **4.3.2. Emission factor retrieval**

For the emission factor retrieval, part following datasets was gathered or generated,

1. Emission factor datasets

This work obtained emission factor datasets for five emission standards (DEFRA, IPCC, CRIS, EPA, and NGA) from their respective official web pages for the years between 2014 to 2021. These files were in pdf, excel, and word formats.

2. Custom word embedding training dataset

Web scrapped a text corpus from Wikipedia pages to train a custom word vector.

3. User satisfaction evaluation dataset

Due to the unavailability of a standard evaluation dataset, work created a custom evaluation dataset with 50 evaluation queries and relevant emission factors to evaluate user satisfaction for different parameters.

#### 4. Personalization training dataset

Since the system is still developing and lacks a proper user history dataset, work generated a synthetic dataset for 50 users.

#### **4.3.3. Unit verification and conversion**

For the implementation of the unit verification model, we will be needing text data consist of different units and measurements of carbon emission detail. In this stage, we are focusing on collecting the datasets from our colleagues using the online survey to collect various carbon emission detail. The collected data set will be having a set of emission factors and their given measurements and units.

Different datasets are required for this measurement verification and unit identification, automate irrigation component, and they must be gathered from outside sources. First, we gathered historical unit calculation products and unit statistics data for the previous 10 years, respectively, for all SI units used globally. Additionally, we have gathered from several colleges the list of advised units for a certain statistical area. Gather measurement information from Google and current unit information from the statics department website.

#### **4.3.4. Emission optimization**

The emission optimization interface of the mobile application will be utilized to collect data on the usage limits of each emission source. After then, optimization algorithms will be used to discover the best possible solution while taking into account the stated limitations. The minimum optimal solution is going to be used to determine the maximum threshold values for each source of emissions. The maximum threshold values will be checked with the business analyst to ensure they meet the requirements of the business. After a confirmation from the business analyst, the threshold values will be saved on the application database. Following that, the threshold values for each source of emission will be compared with each emission from a variety of sources. In



the event that emissions exceeded the limit, an alert will be issued to BA over the email gateway.

#### 4.4. Technologies and Implementation

Technologies and implementation discuss the implementation process handled by each of the four components and list the technologies used by each component. Finally, it shows the application's backend and frontend implementation parts.

##### 4.4.1. Emission activity parts extraction implementation

To build emission extraction part, we used python for the implementation. PyCharm used as the development IDE. For mobile app development used react native. Django used for the backend implementation.

##### 4.4.2. Emission factor retrieval implementation

Table 4.1 shows the technologies and usages of those technologies used in implementing the emission factor retrieval component. It has modularized most functionalities inside python files and used those functionalities in the Jupyter notebooks to visualize, integrate, and test functionalities.

*Table 4.1: EF ranking module implementation technologies and usage*

Technology	Usage
Beautiful Soup	<ul style="list-style-type: none"> <li>• Web scrapping Wikipedia pages for custom word embedding training</li> </ul>
Boto3	<ul style="list-style-type: none"> <li>• For AWS services</li> </ul>
Gensim	<ul style="list-style-type: none"> <li>• Custom word vector pre-processing and training</li> <li>• Obtaining pre-trained word embeddings</li> <li>• Finding embedding similarities</li> </ul>
Joblib	<ul style="list-style-type: none"> <li>• Save TF-IDF matrices as files in the file system</li> </ul>
Jupyter	<ul style="list-style-type: none"> <li>• Research and testing EF ranking system</li> </ul>
MongoDB	<ul style="list-style-type: none"> <li>• NoSQL database for cleaned EF documents, terms lists, and intermediary indexes storage</li> </ul>
NLTK	<ul style="list-style-type: none"> <li>• Language processing tasks, e.g., word tokenization, stop word removal, lemmatization</li> </ul>
NumPy	<ul style="list-style-type: none"> <li>• Linear algebra and vectorized operations</li> </ul>

Pandas	<ul style="list-style-type: none"> <li>• Data preparation</li> <li>• TF-IDF matrix creation</li> <li>• Query scoring</li> <li>• EF ranking</li> </ul>
PyCharm	<ul style="list-style-type: none"> <li>• Research and development IDE</li> </ul>
Pymongo	<ul style="list-style-type: none"> <li>• Connecting and querying to MongoDB within python</li> </ul>
Python	<ul style="list-style-type: none"> <li>• Programming language</li> </ul>
Python-dotenv	<ul style="list-style-type: none"> <li>• Managing credentials in a secure way by storing them as environmental variables</li> </ul>
Scikit-learn	<ul style="list-style-type: none"> <li>• PCA on word vectors for visualization</li> </ul>
Seaborn, matplotlib	<ul style="list-style-type: none"> <li>• Data exploration</li> </ul>
SonarLint	<ul style="list-style-type: none"> <li>• Code quality validation</li> </ul>

#### 4.4.3. Unit verification and conversion implementation

Since I utilize a lot of measurements to calculate my emissions, there are a lot of units here as well. I also need to classify those units.

Before moving on to text categorization, we need a means to quantitatively represent words in a lexicon. Because the majority of our ML models need numbers rather than language.

The one-hot encoding of word vectors is one method to accomplish this, however it is not the best option. This representation would take up a lot of room if there were a large vocabulary, and it would be inaccurate in expressing how similar two words are. For instance, if we wanted to determine the cosine similarity between the numerical words x and y:

The similarity between various words is always going to be 0, given the nature of one-hot encoded vectors. By giving us a fixed-length (often considerably lower than the vocabulary size) vector representation of words, Word2Vec gets around the aforementioned issues. Additionally, it records how various words relate to one another in comparable and similar ways. We may learn many analogies thanks to the manner that word2vec vectors of words are learnt. We can now manipulate words algebraically in ways that were previously not feasible.

The majority of the time, these word vectors have already been trained on sizable text corpora like Wikipedia, Twitter, etc. and are offered by others. Glove and Fast text, which employ 300-dimensional word vectors, are the most often used pre-trained word vectors. We'll be using the Glove word vectors in this post. That's why I utilized this glove to construct my model.

My measurement data is typically not completely accurate. One of the most important processes in the classification pipeline is text preparation since data from various sources have varied properties. The methods we'll discuss in this piece, however, may be used to practically any type of data you would come across in the NLP undergrowth. My preprocessing approach is significantly influenced by the word2vec embeddings that we decide to use for our classification problem. The preprocessing we perform should, in theory, be the same as the preprocessing done before to teaching the word embedding. Since the majority of embeddings don't provide vector values for punctuation and other special characters, I first want to remove them from my text data.

Deep learning algorithms take a sequence of text as input and understand its structure exactly like humans do, which is one factor that has made deep learning a top choice for NLP. We no longer need to manually create features from our text data. Machines want data in numerical form because they cannot interpret speech. Consequently, we must display our text data as a collection of numbers. We must have a basic understanding of the Keras Tokenizer function in order to comprehend how this is accomplished. Although there are other potential tokenizers, I think the Keras Tokenizer is a decent option.

My model typically anticipates that each text sequence (training example) will have a same length (number of words/tokens). With the help of the maxlen option, I can manage this. My training data currently consists of a list of numbers. The length of each list is the same. Additionally, I have the word index, a dictionary of the terms used most frequently in the corpus of text.

The Pytorch model anticipates a number rather than a string for the target variable. Our target variable may be converted using the Label encoder from Sklearn. Since I had so many targets at this time, I used many classifications. First, I loaded the necessary glove embeddings.

The folder where you downloaded these GLOVE vectors must be specified. The embeddings index is an array of length 300 that contains a dictionary with the word as the key and the word vector as the value. This dictionary is around one billion words long.

#### **4.4.4. Emission optimization implementation**

The following technologies have been used to implement carbon emission optimization module.

- Python - Programming language which was used to implement optimization algorithm.
- PyCharm - Research and development IDE
- Pyomo - Pyomo is a software program that is available under the open-source license that used to formulate and solve large-scale optimization problems.
- NumPy - NumPy is a scientific computing package. It is a core library that offers great speed and tools for array objects.
- Pandas - The Panda library is an open-source library that facilitates data analysis and is simple to use. It provides a data structure with good performance and ease of use.
- GLPK - The GNU Linear Programming Kit (GLPK) solver is an open-source tool that was developed to tackle linear programming, mixed-integer, and other problems that are linked to these types of programming.
- Django - Django is a high-level web framework written in Python that enables the rapid building of websites that are both safe and easy to maintain which is used to backend implementation of research project.
- React Native - Building native mobile applications with JavaScript is made possible with the help of React Native, which is a framework that is used for the front-end mobile application development.

- Mongo DB - NoSQL database for cleaned EF documents, terms lists, and intermediary indexes storage.

#### 4.4.5. Application backend implementation

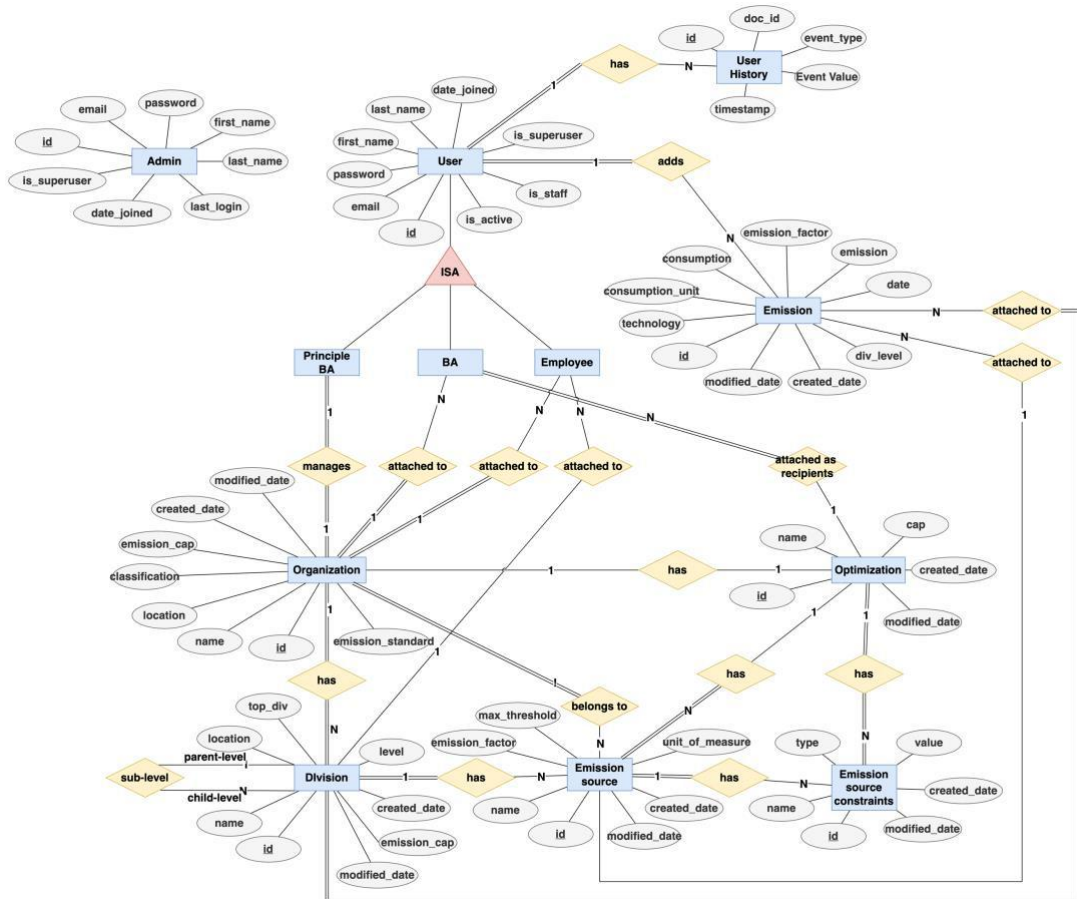


Figure 4-5: Application database Entity Relation (ER) diagram

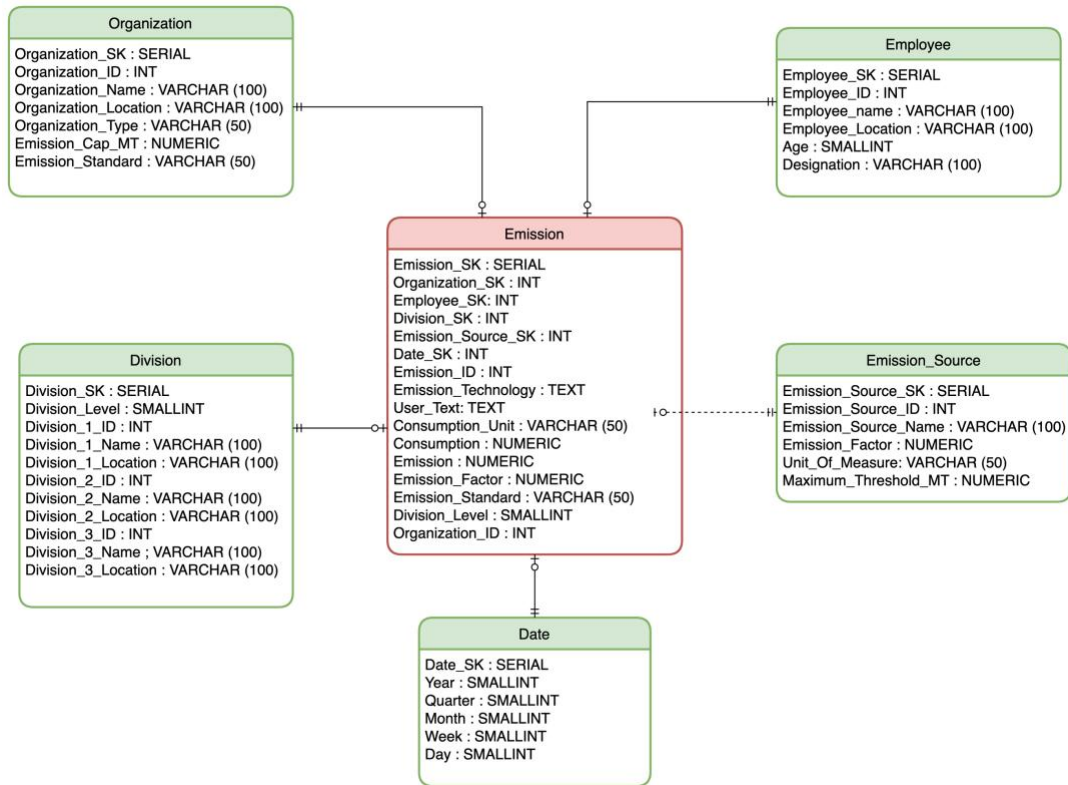


Figure 4-6: Data warehouse physical diagram

Initially, the backend development designed the database entity relationship diagram (shown in Figure 4-5), data warehouse design (shown in Figure 4-6), and API endpoints. Table 4.2 shows the technologies used in developing and testing the application's backend.

Table 4.2: Backend implementation technologies and usage

Technology	Usage
Django	<ul style="list-style-type: none"> <li>Backend framework</li> </ul>
Django-rest	<ul style="list-style-type: none"> <li>Representational State Transfer (REST) API capabilities</li> </ul>
PostgreSQL	<ul style="list-style-type: none"> <li>Application database and data warehouse</li> </ul>
Postman	<ul style="list-style-type: none"> <li>API endpoint testing</li> </ul>
PyCharm	<ul style="list-style-type: none"> <li>Development IDE</li> </ul>
Python	<ul style="list-style-type: none"> <li>Programming language</li> </ul>
Python-dotenv	<ul style="list-style-type: none"> <li>Managing credentials in a secure way by storing them as environmental variables</li> </ul>
SonarLint	<ul style="list-style-type: none"> <li>Code quality validation</li> </ul>

#### 4.4.6. Application frontend implementation

Withing application frontend development, initially designed UI prototypes (shown in Appendix A. UI Prototypes) and finally developed and tested the frontend mobile app using the technologies shown in Table 4.3.

Table 4.3: Mobile frontend development technologies and usage

Technology	Usage
Expo-CLI	<ul style="list-style-type: none"><li>• Representational State Transfer (REST) API capabilities</li></ul>
Figma	<ul style="list-style-type: none"><li>• User Interface (UI) prototyping</li></ul>
JavaScript	<ul style="list-style-type: none"><li>• Programming language</li></ul>
WebStorm	<ul style="list-style-type: none"><li>• Development IDE</li></ul>
React-Native	<ul style="list-style-type: none"><li>• Cross-platform mobile application development</li></ul>
SonarLint	<ul style="list-style-type: none"><li>• Code quality validation</li></ul>

#### 4.5. Experimentations

Experimentation sections show the type of evaluation experiments accomplished by each component to evaluate the success metrics of each component implementation.

##### 4.5.1. Emission factor retrieval evaluation

The emission factor retrieval component's evaluation included the following evaluation criteria and relevant methodologies,

1. User satisfaction

**Motivation:** user satisfaction is an essential measure in evaluating search features. Several metrics are available to evaluate user satisfaction; however, the used Mean Average Precision (MAP) is the single measure.

**Experimentation methodology:** used the evaluation dataset generated with the implemented system to evaluate MAP for delta values and word embeddings. Found the best MAP values using a surrogate model called the Gaussian process, which converges at the maximum by iterating the evaluation for different parameter values.

2. Query speed

**Motivation:** query speed (or time taken) for a single query execution also affects usability.

**Experimentation methodology:** measured average CPU time for 250 queries of different standard documents.

3. Emission-factor scalability

**Motivation:** the scalability of the emission factor search system highly depends on the ability of the architecture to support new emission factor standards.

**Experimentation methodology:** measured time (human hours) to adopt a new emission factor standard (IPCC).

4. System resource utilization

**Motivation:** Having a lower system resource utilization can benefit the system's scalability and cost management.

**Experimentation methodology:** experimentation measured memory and storage usage by the system components.

#### 4.5.2. Emission optimization evaluation

The emission optimizations component's evaluation included the following evaluation criteria and relevant methodologies,

1. Find threshold for each emission source.

**Motivation:** Find the optimum solution for each emission source.

**Experimentation methodology:** Using linear programming algorithms we have done the experiment to find maximum threshold value to each emission source.

2. Sent an alert for any violation of threshold.

**Motivation:** Reduce the emission within the given carbon credit.

**Experimentation methodology:** Using linear programming algorithms we have done the experiment to find maximum threshold value to each emission source.

3. Optimization scalability

**Motivation:** Applications should be scalable in accordance with the expansion of the organization.



Experimentation methodology: Experiment have done for multiple optimization which was create by BA and optimization have done it for multiple sources.

#### 4.6. Commercialization

This section discusses the commercialization steps and future marketing strategies to promote this system. Completed commercialization techniques as follows,

1. Market analysis

A market study was conducted for the system (Carbonis) to find the high usability demand for emission calculation tools.

2. Creating a business model

Figure 4-7 shows the business model canvas illustrating the system's business model.

3. Creating a pricing plan

Figure 4-8 illustrates the current pricing plans according to the business model canvas.

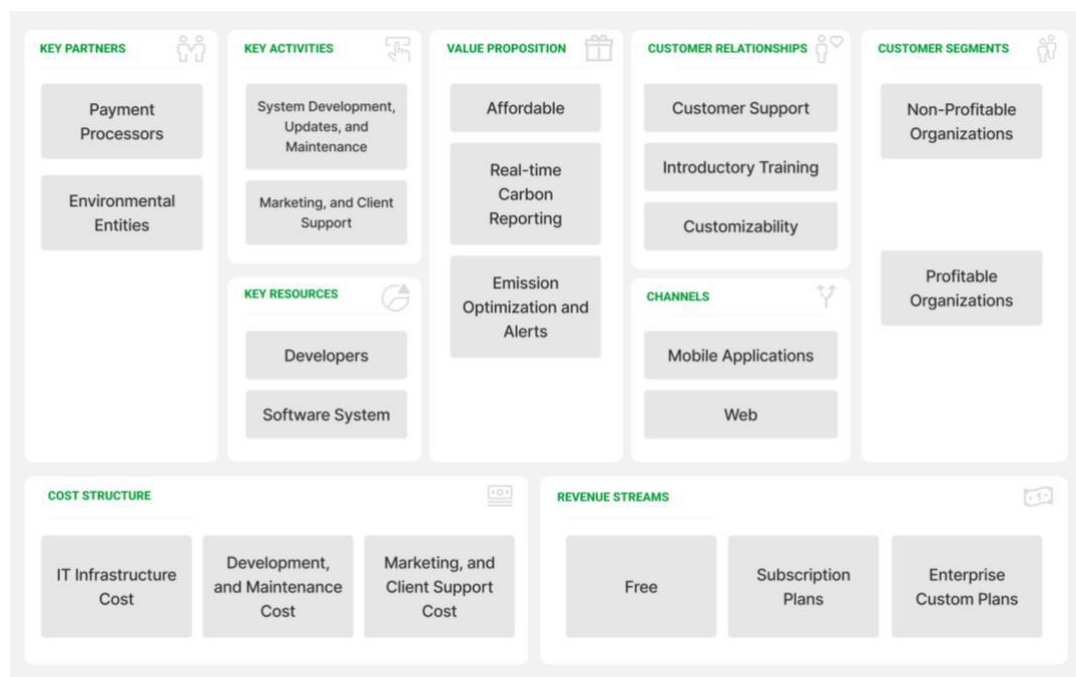


Figure 4-7: Business model canvas



Figure 4-8: Pricing plan

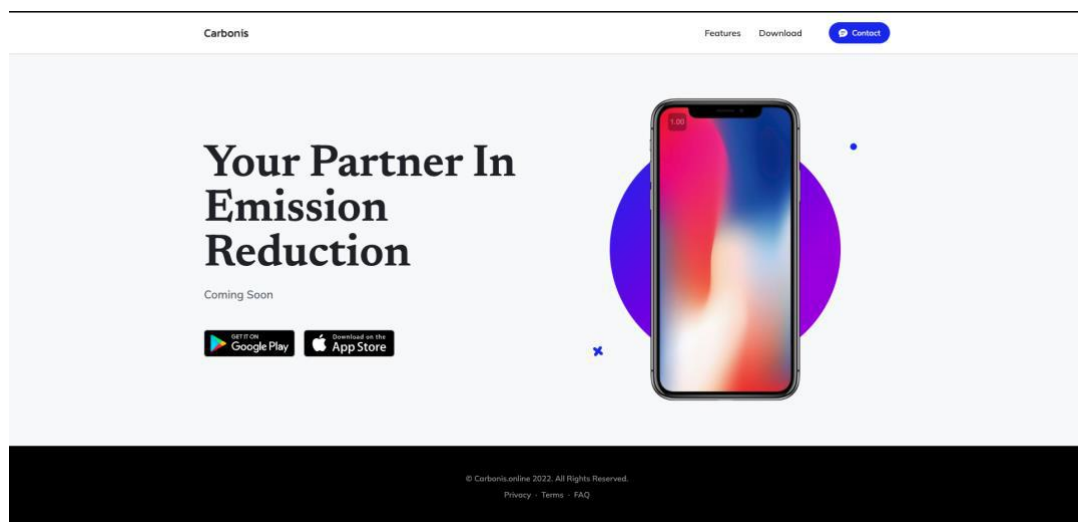


Figure 4-9: Product landing page

#### 4. Registering a professional domain address

Registered a web domain address (“www.carbonis.online”) for the landing and app deployment.

#### 5. Creating a product landing page

Figure 4-9 shows the currently created product landing page deployed.

#### 6. Designing a product pamphlet

As shown in Figure 4-10, designed a promotion pamphlet.

Future commercialization tasks are as follows,

1. Search Engine Optimization (SEO) for the landing page
2. Social media promotion
3. Creating a demo video
4. Creating a commercial advertisement video
5. Distributing promotional pamphlets



Figure 4-10: Product promotion pamphlet

## 5. RESULTS AND DISCUSSION

The results and discussion section presents the results of each research component and discusses the findings of each corresponding component.

### 5.1. Results

#### 5.1.1. Emission activity parts extraction results

User satisfaction: We tested this component with some random groups. Below is the result we gathered from them. Even this NER worked well. When it is comes to normal users, they need some basic training about this app. In below picture shows that, nearly 60% users feel good about this application.

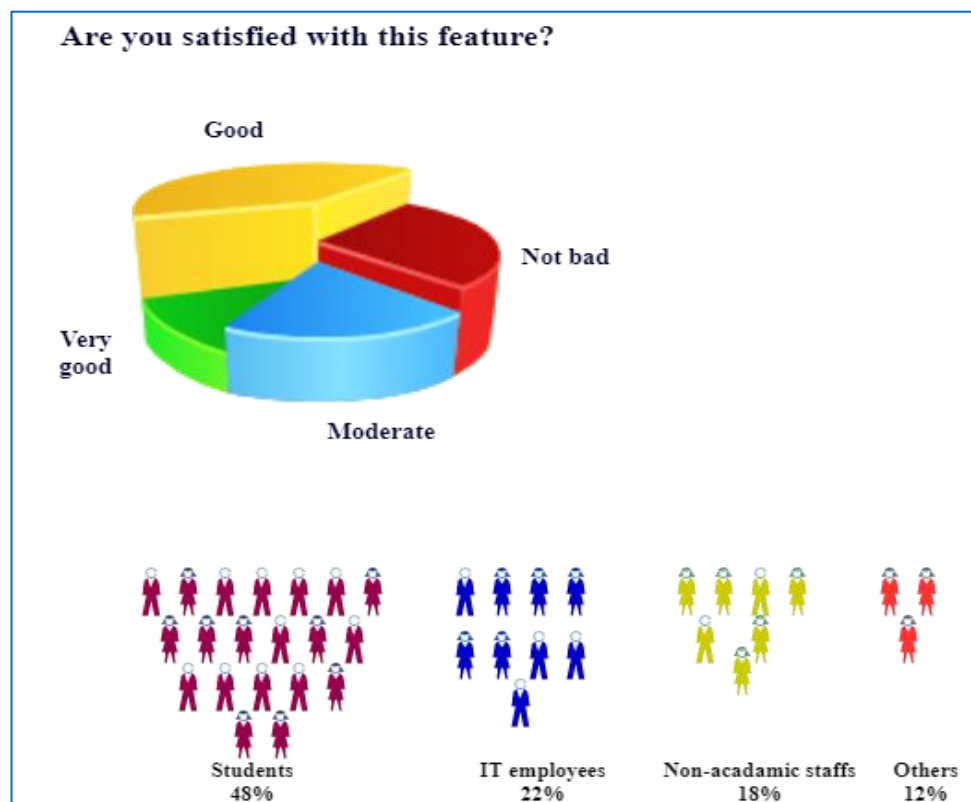


Figure 5-1: User satisfaction result

Average speed: Even When it is comes to speed, for a mobile application speed is one of the most important things. The application should be able to load fast. We gave this application to some set of users and the below graph shows the speed of the application pages and the loading speed.

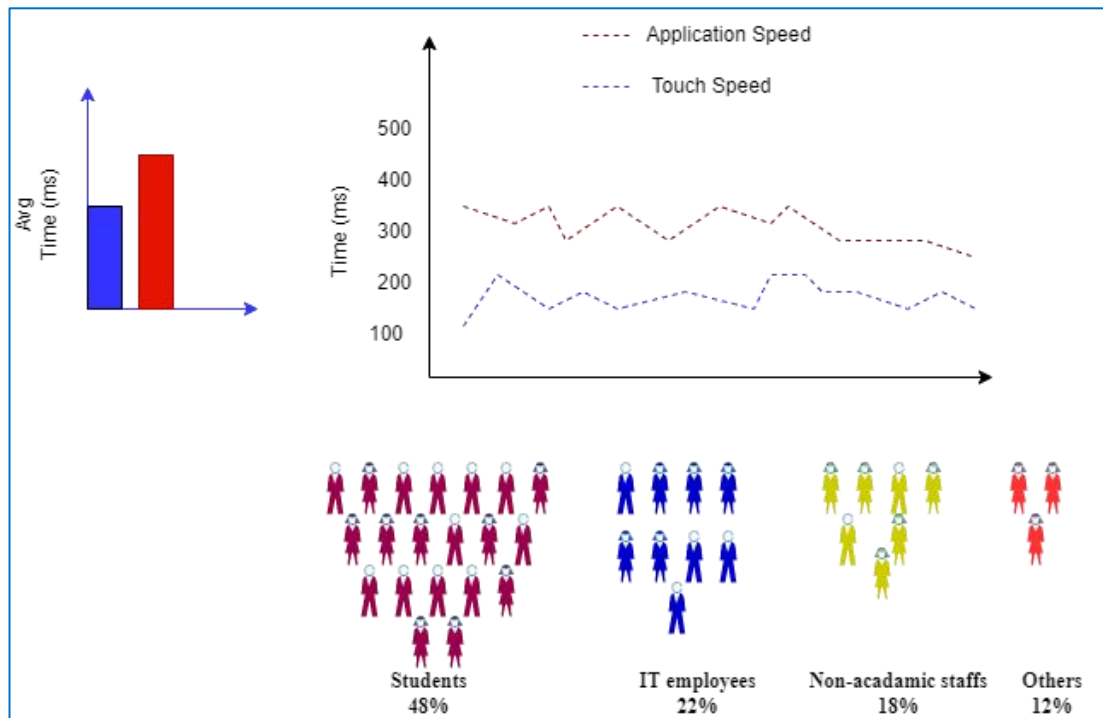


Figure 5-2: Application and Extraction average speed

### 5.1.2. Emission factor retrieval results

**User satisfaction:** in the automated optimization, optimization algorithms from surrogate models, such as the gaussian process, tree-based search, and grid search, worked successfully. Among these, the Gaussian process provided the best possible findings within the shortest time or iterations. Figure 5.1 shows the results of such optimization

**Average query speed:** Figure 5-4 and Figure 5-5 illustrates average time results with a different term and EF document counts. Figure 5-6 shows the average times of various word vectors and delta values. In addition, DEFRA and IPCC had average times of nearly 170 and 540 milliseconds, respectively. Table 5.1 shows the average times for EF datasets with the highest EF document and terms counts.

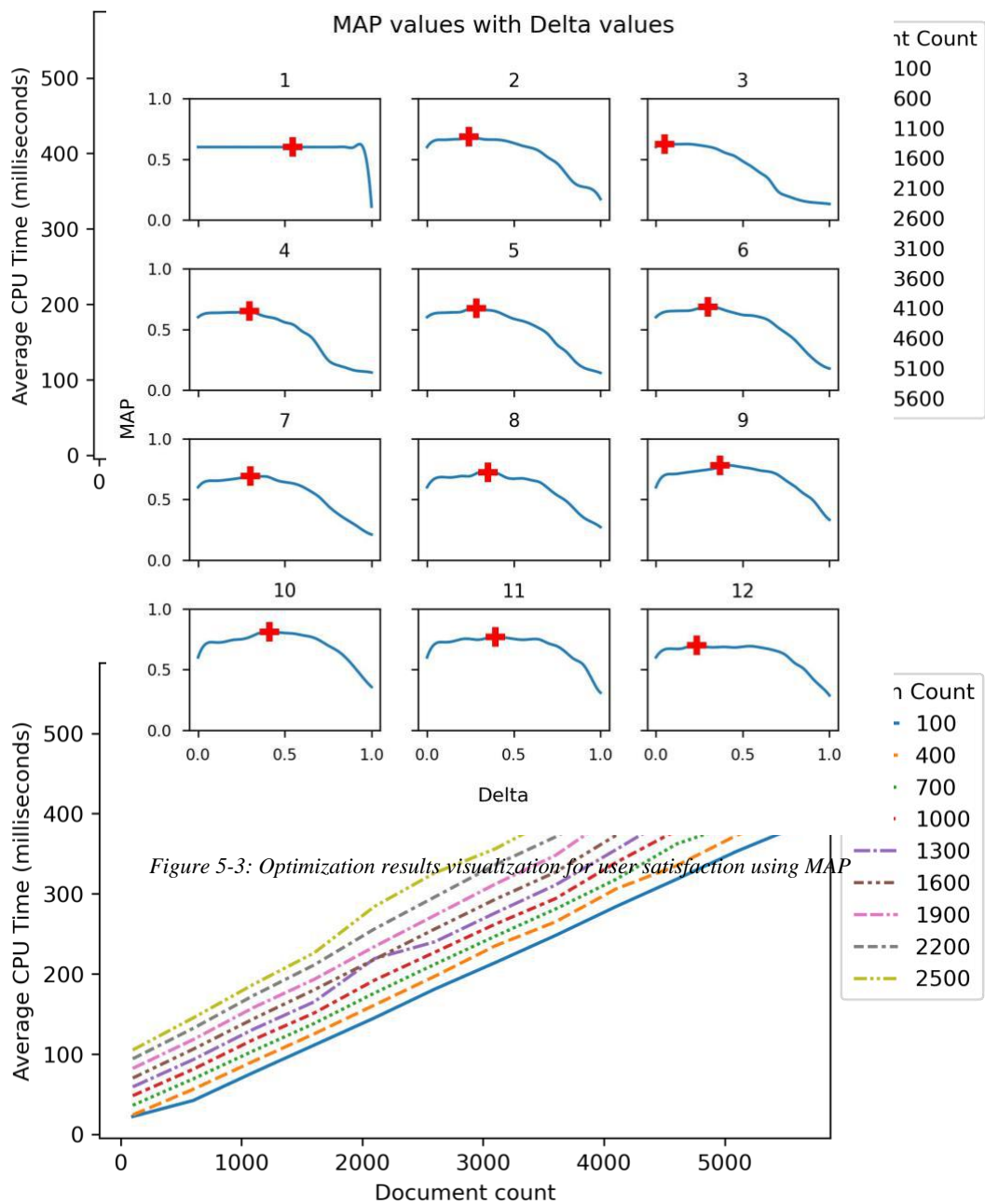


Figure 5-5: Average CPU time with document count

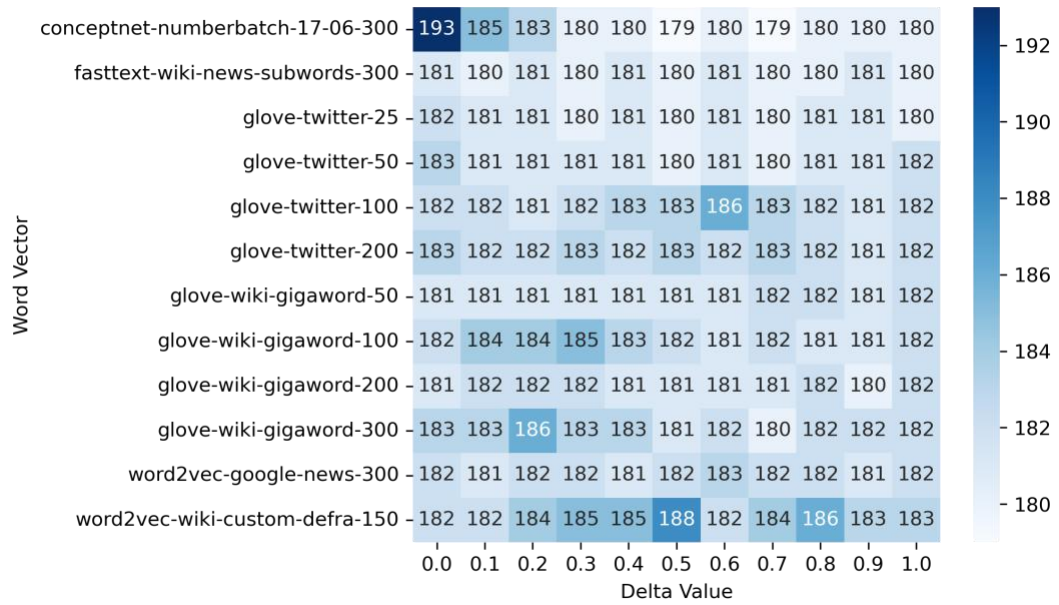


Figure 5-6: Average CPU time with word vector and delta values

Table 5.1: Average query time for vast EF datasets

EF Standard	EF Document Count	Term Count	Average Time (milliseconds)
DEFRA 2021	2435	365	181
IPCC 2006	6201	2508	536

**EF scalability:** Adoption of IPCC (1996 and 2006) took approximately 12 person-hours with no code modification except the PC1 part. However, PC1 is supposed to have some changes during the new EF adoption.

**System resource utilization:** Table 5.2 shows the memory and storage usage of TF-IDF matrices. TF-IDF matrices use the same amount of storage and memory.

Table 5.2: Memory and storage usage by TF-IDF matrices

EF Standards	Year	Matrix Size (document count, term count)	Memory or Storage Usage (MB)	Total Standard Usage (MB)
DEFRA	2014	1953, 332	5.2	50.4
	2015	2015, 361	5.8	

	2016	1989, 336	5.4	
	2017	2353, 346	6.5s	
	2018	2389, 353	6.8	
	2019	2390, 354	6.8	
	2020	2367, 357	6.8	
	2021	2435, 365	7.1	
IPCC	1996	6201, 2508	124.5	248.9
	2006	6201, 2508	124.5	

### 5.1.3. Emission optimization results

The findings of the research were summarized into three primary categories after being analyzed. System validity, carbon control, and the organizational structure of the business.

The blueprint for the system was well built so that it could successfully accomplish the necessary calculation and optimization of carbon emissions. The approach employed an algorithm that was unobtrusive enough to calculate the carbon emission and offer results in real time that had very accurate numerical values. In addition, both the front-end and back-end processes are formed by following the appropriate patterns and are accurately monitored by the user as well as the developer. In addition, because we stored the data on a cloud server, the real-time outputs were much simpler to handle and were much more sensitive to minute adjustments in the calibration. This results in a significant reduction in the amount of time needed to load the system and simplifies the overall experience for the user. Validation was performed using the appropriate settings and values for the current time period on the results and outcomes of the carbon neutrality tests as well as the data provided for optimizations.

The technique proved quite fruitful in terms of controlling the carbon emission produced by the organizations. The algorithm was designed to gradually lower the threshold by a smaller and smaller amount over the course of time. Consequently, it was graphed that the organizations' carbon emission will decrease with time in the next years. This results in the organization using fewer carbon credits overall, and it also reduces the amount of carbon that it emits. This leads to a reduction in the amount of



carbon that is consumed by a certain percentage, and once the source of carbon emissions reaches its threshold, the system appears to follow an intelligent and sustainable path to function with no emissions of carbon due to the fact that carbon credits are reserved for a variety of important sources.

The research, on the other hand, is supposed to concentrate on the management of real-time carbon neutrality and the optimization of climate circumstances; moreover, the central structure should produce a clean and successful business scheme that we may develop further. As a result, the cloud server storage and algorithm have the possibility of creating a business framework for the management of carbon control in organizations. With simply under the surveillance of BA, the entire carbon neutrality will be done and managed for the businesses, which boosts the efficiency and carbon waste management of the organization while simultaneously requiring less human labor. Because the machine language system provides an extremely accurate response to the optimization of real-time calculations, we are able to alter it to construct a comprehensive application geared toward the needs of businesses and huge institutions that generate a lot of traffic and use a lot of carbon. As a result of this research, a straightforward and easily digestible essential business structure in the field of carbon neutrality management has been produced.

## **5.2. Research Findings**

### **5.2.1. Emission factor retrieval finding**

#### **User satisfaction:**

- **Best word vector:** glove-wiki-gigaword-300 outperformed others with a MAP of 0.81, nearly 30% improvement over plane VSM (0.60 MAP), and 127% improvement over plane word embedding (0.36 MAP). glove-wiki-gigaword (100, 200) and word2vec-google-news-300 also performed well.
- **Best delta value:** Even though the best delta range changes with word vector, most word vectors performed well between the 0.35 and 0.5 delta range, and glove-wiki-gigaword-300 gave the highest MAP at 0.41 delta value.

- **Improvement by combined rank approach:** With the combined rank approach, almost all word vectors improved MAP except conceptnet-numberbatch-17-06-300 (no improvement for any delta) over plane VSM or Embedding scoring.

#### **Average query speed:**

- **Effect of EF document and term counts:** Average time increases with EF document and terms counts. However, even the vast EF datasets of DEFRA and IPCC only had sub-second average times. Therefore, the average speed can be considered acceptable for current EF datasets.

#### **EF scalability:**

- **EF scalability:** Current EF retrieval system adopts new EF standards and datasets with minimal effort and code changes.

#### **System resource utilization:**

- **The best scalable word vector:** glove-wiki-gigaword-300, wins with low resource utilization considering its high user satisfaction. Therefore, glove-wiki-gigaword-300 can be considered the best scalable word vector among these word vectors.

#### **5.2.2. Unit verification and conversion finding**

The project's basic concept is to create an Android application that may assist corporations in choosing carbon-free crops. Many raw datasets that have an impact on the emission activities, such as transportation data, electricity readings, and others, should be used to construct the detect model.

For all machine learning and deep learning models, model training is crucial; algorithm performance depends on dataset learning. Since we are inexperienced researchers, the algorithms' initial performance was poor. To learn how to improve performance, we read a lot of study papers and publications.

### **5.2.3. Emission optimization finding**

capable of determining the optimal threshold for every emission source. An warning should be sent to the BA in real time if the emission exceeds the ideal solution. Consequently, the organizations should control their emission operations without going over their allotted emission credits. In order for the company to meet its emission goals and become carbon neutral, carbon emission optimization will be very beneficial.

## **5.3. Discussion**

### **Emission activity parts extraction:**

In this Emission parts extraction - As per our problem, our requirement is to extract the emission factors from the employees' text/input. In this paper we gave several solutions/models to solve. But only some models worked well for our requirement. Three different measures calculated for each NER models. The best model selected based on the performance. However, with more data we can build a most accurate model so that we can get emission activities from any kind of (with spelling errors) sentences/inputs.

**Emission factor retrieval:** The EF retrieval system's combined ranking approach showed improved user satisfaction over plane VSM and word embedding. As explained in the word embedding preliminary, corpus domain, corpus size, dimension, and algorithms affect user satisfaction, and Wikipedia corpus, GloVe, or word2vec algorithm, decent corpus size and dimension provides better user satisfaction. Query speed is acceptable and decreases with EF document and term counts. EF retrieval system scales for new EF standards, datasets, and user space. The gaussian process can successfully automate finding the best word vectors and delta parameter values. The AWS personalize trained models could further improve the rankings.

**Emission optimization:** In the future, we plan to gather more data, particularly for the unconstrained CO<sub>2</sub> model, so that we may enhance the evaluation of the constrained models and possibly incorporate it during the training. This will allow us to collect more data for the unconstrained CO<sub>2</sub> model. Specifically, our attempts to collect data

will focus the majority of their attention on the leaf diseases that are not limited in any way. In order to get high accuracy, we might employ a variety of optimization approaches.

Even though we want to collect more data in an unrestricted context, information gathered in a restricted environment will still constitute the vast majority of our total data. As a consequence of this, one of our objectives is to enhance our grasp of the impact that the constraints imposed on the data collection have had on the generalization performance. If we take these actions, we will be able to improve the process of data collection, which will enable us to lessen the impact of any negative repercussions that may occur. In order to make the most of the chances afforded by the restricted data collection method, we plan to publish an app that will make it possible to collect data through the use of crowd sourcing.

#### 5.4. Summary of Each Student Contribution

Student	Role	Component	Contributions
Sathees P (IT19052748)	Team leader	Emission factor retrieval	<ul style="list-style-type: none"> <li>• Emission factor retrieval's research, development, testing, and integration</li> <li>• Designs: database, data warehouse, API endpoints, and UI prototypes</li> <li>• Database creation and deployment</li> <li>• Datawarehouse creation and deployment</li> <li>• Backend API endpoints creation (nearly 50 endpoints) and testing, including emission factor retrieval</li> <li>• Data warehousing function</li> <li>• Backend deployment to AWS</li> <li>• Frontend project setup, emission flow creation, emission factor retrieval, navigation, viewing emissions,</li> </ul>

			<p>authentication, and authorization and testing</p> <ul style="list-style-type: none"> <li>• Commercialization: market analysis, business model canvas, pricing plan, landing page, and pamphlet creation</li> <li>• Project management and team coordination</li> <li>• Individual research paper publication</li> <li>• Documentation tasks</li> </ul>
Mathanika M (IT19005218)	Member	Emission activity parts extraction	<ul style="list-style-type: none"> <li>• The Emission activity parts extraction research, development, testing, and integration</li> <li>• Emission activity parts extraction's API endpoint creation and testing</li> <li>• Emission activity parts extraction's frontend development and testing</li> <li>• Commercialization: market analysis</li> <li>• Combined research paper publication</li> <li>• Documentation tasks</li> </ul>
Vishakanan S (IT19001562)	Member	Unit verification and conversion	<ul style="list-style-type: none"> <li>• Unit verification and conversion research, development, testing, and integration</li> <li>• Unit verification and conversion endpoint integration and testing</li> <li>• Login frontend UI development</li> <li>• Commercialization: market analysis</li> <li>• Combined research paper publication</li> <li>• Documentation tasks</li> </ul>
Vithursan M (IT19033174)	Member	Emission optimization	<ul style="list-style-type: none"> <li>• Emission optimization's research, development, testing, and integration</li> </ul>

			<ul style="list-style-type: none"> <li>• Emission optimization's API endpoint creation and testing</li> <li>• Emission optimization's frontend development and testing</li> <li>• Commercialization: pricing plan</li> <li>• Individual research paper publication</li> <li>• Documentation tasks</li> </ul>
--	--	--	--

## 6. CONCLUSIONS

**Emission activity parts extraction:** Maintaining carbon neutrality is the most crucial factor in today's globe. If we refuse, we shall reach a critical juncture. As a result, all governments should take action to make the planet a healthier place, every nation should set limits on carbon emissions of every organization. Our proposed system will be quite useful in solving this problem. Every organization can calculate their daily emissions and regulate the number of emissions within their control utilizing our technology

**Emission factor retrieval:** This component research proves that the combined ranking approach will improve usability and be scalable for EF selection tasks compared to plane VSM or word embedding. It is possible to improve the usability and scalability of the emission calculation interface with this EF retrieval system. Can expect to save user time and effort during emission calculation. An improvement of 29.95% and 127% was observed over VSM and embedding rankings, respectively, by the combined rank approach with glove-wiki-gigaword-300 for a 0.41 linear combination parameter (delta) value. All queries took a sub-second delay for DEFRA (2014 to 2021) and IPCC (1996 and 2006). Similar surrogate optimization methods could automate ranking evaluation. However, further experimentations and improvements, such as performance and memory optimizations, are needed. Future tasks include scaling other EF standards and experimenting with this retrieval system. Additionally, the plan includes applying heuristic optimizations to improve performance and scalability.

**Unit verification and conversion:** In conclusion, this study classifies units using the TensorFlow deep learning framework. Three of the three goals of this study have been met over its course. The aims and conclusions are closely tied to one another since it may have an impact on whether all of the objectives are successfully reached. It may be claimed that all of the studies had very exceptional outcomes. The main topic of this study is the recurrent neural network (RNN), notably in text categorization technology. We looked more closely at RNN technology, starting with model

construction, training, and the division of units into measures. Epochs in RNNs have the ability to control accuracy and prevent problems like overfitting.

TensorFlow, a framework for implementing deep learning, produced positive findings as well since it can simulate, train, and classify data with up to 80% accuracy for numerous measurements that have been turned into trained models. Finally, Python has been employed throughout this research as the programming language since it is compatible with the TensorFlow framework, which enables Python to be used throughout the whole system design process.

**Emission optimization:** The production of excessive amounts of carbon at this period of rapid modernization plays an essential part in the fight against climate change. The path that leads to a society that is more environmentally friendly is paved with the pursuit of carbon neutrality and the experimentation of environmentally responsible methods of its management. It was seen that there was a need for greater study in the application of carbon reporting at a corporate level, and the majority of the research that is now accessible focuses on the efficiency of various reporting schemes. The significance of this research lies in the fact that it seeks to improve carbon efficiency in order to conserve energy and lower undesirable emissions. Implementing a real-time platform that is able to provide insights into the organization's most recent emission statistics is one creative approach that has been proposed as a response to the issue that was described above. Assessing a company's carbon footprint is often the first step that businesses take on the path to becoming carbon neutral. By doing so, they will be able to determine the most effective strategic strategy to reducing emissions and setting lofty targets toward reaching carbon neutrality. The research was fruitful as it was written; nevertheless, there are a few recommendations that will make the research that comes after this paper more straightforward. One of the drawbacks of our research was that the user-provided data, such as limits and ranges, needed to be correctly supplied for the algorithm to respond to real-time improvement. The real-time technique was pretty precise to conventional optimization, which is the ideal practice that we propose for you to follow; however, it needs to be further enhanced for higher institutes while collecting data in bulk. To summarize, emission optimization for real-time carbon neutrality management can be carried out in a



manner that is both very efficient and productive in the context of carbon control in companies.

## REFERENCES

- [1] M. Roelfsema *et al.*, “Taking stock of national climate policies to evaluate implementation of the Paris Agreement,” *Nature Communications* 2020 11:1, vol. 11, no. 1, pp. 1–12, Apr. 2020, doi: 10.1038/s41467-020-15414-6.
- [2] J. Bebbington, C. Larrinaga-González, C. Larrinaga-Gonzálezgonza’, and G. Ãã, “Carbon Trading: Accounting and Reporting Issues,” <http://dx.doi.org/10.1080/09638180802489162>, vol. 17, no. 4, pp. 697–717, 2008, doi: 10.1080/09638180802489162.
- [3] F. Schreyer *et al.*, “Common but differentiated leadership: strategies and challenges for carbon neutrality by 2050 across industrialized economies,” *Environmental Research Letters*, vol. 15, no. 11, p. 114016, Oct. 2020, doi: 10.1088/1748-9326/ABB852.
- [4] K. Piper, J. Longhurst, M. Khare, and Z. P. Robinson, “Exploring corporate engagement with carbon management techniques,” *Emerald Open Research* 2021 3:9, vol. 3, p. 9, May 2021, doi: 10.35241/emeraldopenres.14024.1.
- [5] C. Quitmann, R. Sauerborn, and A. Herrmann, “Gaps in Reporting Greenhouse Gas Emissions by German Hospitals—A Systematic Grey Literature Review,” *Sustainability* 2021, Vol. 13, Page 1430, vol. 13, no. 3, p. 1430, Jan. 2021, doi: 10.3390/SU13031430.
- [6] “SRI LANKA UPDATED NATIONALLY DETERMINED CONTRIBUTIONS”.
- [7] H. Hashim *et al.*, “An Integrated Carbon Accounting and Mitigation Framework for Greening the Industry,” *Energy Procedia*, vol. 75, pp. 2993–2998, Aug. 2015, doi: 10.1016/J.EGYPRO.2015.07.609.
- [8] B. Sarkar, M. Omair, and S. B. Choi, “A Multi-Objective Optimization of Energy, Economic, and Carbon Emission in a Production Model under Sustainable Supply Chain Management,” *Applied Sciences* 2018, Vol. 8, Page 1744, vol. 8, no. 10, p. 1744, Sep. 2018, doi: 10.3390/APP8101744.
- [9] S. Tang and D. Demeritt, “Climate Change and Mandatory Carbon Reporting: Impacts on Business Process and Performance,” *Bus Strategy Environ*, vol. 27, no. 4, pp. 437–455, May 2018, doi: 10.1002/BSE.1985.

- [10] C. L. Spash, "The Brave New World of Carbon Trading," 2009.
- [11] F. Wang *et al.*, "Technologies and perspectives for achieving carbon neutrality," *The Innovation*, vol. 2, no. 4, p. 100180, Nov. 2021, doi: 10.1016/J.XINN.2021.100180.
- [12] A. A. Rahman, Y. A. Aziz, and S. Sidek, "A Review on Drivers and Barriers towards Sustainable Supply Chain Practices Readiness of Malaysian Food-based Logistics Service Providers for Halal Practices View project Impact of food quality, food image and perceive value on tourist satisfaction and behavioral intentions. View project", doi: 10.7763/IJSSH.2015.V5.575.
- [13] M. Brander, M. Gillenwater, and F. Ascui, "Creative accounting: A critical perspective on the market-based method for reporting purchased electricity (scope 2) emissions," *Energy Policy*, vol. 112, pp. 29–33, Jan. 2018, doi: 10.1016/J.ENPOL.2017.09.051.
- [14] G. Shil, "Greenhouse Gas Emissions: A Case Study of Development of Data Collection Tool and Calculation of Emissions Outline GHG Emissions Inventory Development Process Reporting principles Organizational and operational boundaries Data collection and validation," 2007.
- [15] "Government conversion factors for company reporting of greenhouse gas emissions - GOV.UK." <https://www.gov.uk/government/collections/government-conversion-factors-for-company-reporting> (accessed Jan. 24, 2022).
- [16] "CRIS Public Reports | The Climate Registry." <https://www.theclimateregistry.org/our-members/cris-public-reports/> (accessed Jan. 24, 2022).
- [17] "GHG Emission Factors Hub | US EPA." <https://www.epa.gov/climateleadership/ghg-emission-factors-hub> (accessed Jan. 24, 2022).
- [18] "National Greenhouse Accounts Factors | Department of Industry, Science, Energy and Resources." <https://www.industry.gov.au/data-and-publications/national-greenhouse-accounts-factors> (accessed Jan. 24, 2022).
- [19] R. D. S. Jayathunga and M. H. N. K. T. Dulani, "A GUIDE for CARBON FOOTPRINT ASSESSMENT CLIMATE CHANGE SECRETARIAT

MINISTRY OF MAHAWELI DEVELOPMENT AND ENVIRONMENT The Climate Change Secretariat Ministry of Mahaweli Development and Environment,” 2016.

- [20] “CarbonView – Carbon reporting made easy.” <https://carbon-view.com/> (accessed Jan. 24, 2022).
- [21] “Simplified Carbon Reporting with Turbo Carbon™ | UL.” <https://www.ul.com/services/digital-applications/simplified-co2-reporting> (accessed Jan. 24, 2022).
- [22] “Carbon Management & Reporting - Sphera.” <https://sphera.com/carbon-management-reporting/> (accessed Jan. 24, 2022).
- [23] B. Tranberg, O. Corradi, B. Lajoie, T. Gibon, I. Staffell, and G. B. Andresen, “Real-time carbon accounting method for the European electricity markets,” *Energy Strategy Reviews*, vol. 26, p. 100367, Nov. 2019, doi: 10.1016/J.ESR.2019.100367.
- [24] M. Roelfsema *et al.*, “Taking stock of national climate policies to evaluate implementation of the Paris Agreement,” *Nature Communications* 2020 11:1, vol. 11, no. 1, pp. 1–12, Apr. 2020, doi: 10.1038/s41467-020-15414-6.
- [25] “Measuring and reporting environmental impacts: guidance for businesses - GOV.UK.” <https://www.gov.uk/guidance/measuring-and-reporting-environmental-impacts-guidance-for-businesses> (accessed Oct. 18, 2022).
- [26] “Guidance on how to measure and report your greenhouse gas emissions,” 2009, Accessed: Oct. 18, 2022. [Online]. Available: [www.defra.gov.uk](http://www.defra.gov.uk)
- [27] M. Bhavadharani, M. P. Ramkumar, and S. G. S. R. Emil, “Performance analysis of ranking models in information retrieval,” *Proceedings of the International Conference on Trends in Electronics and Informatics, ICOEI 2019*, pp. 1207–1211, Apr. 2019, doi: 10.1109/ICOEI.2019.8862785.
- [28] L. Xiaoli, Y. Xiaokai, and L. Kan, “An improved model of document retrieval efficiency based on information theory,” *J Phys Conf Ser*, vol. 1848, no. 1, p. 012094, Apr. 2021, doi: 10.1088/1742-6596/1848/1/012094.
- [29] M. Farouk, “Measuring text similarity based on structure and word embedding,” *Cogn Syst Res*, vol. 63, pp. 1–10, Oct. 2020, doi: 10.1016/J.COGSYS.2020.04.002.

- [30] S. Lai, K. Liu, S. He, and J. Zhao, “How to generate a good word embedding,” *IEEE Intell Syst*, vol. 31, no. 6, pp. 5–14, Nov. 2016, doi: 10.1109/MIS.2016.45.
- [31] Y. Y. Lee, H. Ke, T. Y. Yen, H. H. Huang, and H. H. Chen, “Combining and learning word embedding with WordNet for semantic relatedness and similarity measurement,” *J Assoc Inf Sci Technol*, vol. 71, no. 6, pp. 657–670, Jun. 2020, doi: 10.1002/ASI.24289.
- [32] B. Wang, A. Wang, F. Chen, Y. Wang, and C. C. J. Kuo, “Evaluating word embedding models: methods and experimental results,” *APSIPA Trans Signal Inf Process*, vol. 8, pp. 1–14, 2019, doi: 10.1017/ATSIP.2019.12.
- [33] X. Yang, D. Lo, X. Xia, L. Bao, and J. Sun, “Combining Word Embedding with Information Retrieval to Recommend Similar Bug Reports,” *Proceedings - International Symposium on Software Reliability Engineering, ISSRE*, pp. 127–137, Dec. 2016, doi: 10.1109/ISSRE.2016.33.

# APPENDICES

## Appendix A. UI Prototypes

