Real-time Carbon Neutrality Management and Optimization Using Natural Language Processing

Project ID – 2022-175

S.Vishakanan

(IT19001562)

Bachelor of Science (Hons) in Information Technology

Specializing in Data Science

Department of Information Technology

Sri Lanka Institute of Software Engineering

October 2022

Real-time Carbon Neutrality Management and Optimization Using Natural Language Processing

Project ID – 2022-175

S.Vishakanan

(IT19001562)

Bachelor of Science (Hons) in Information Technology

Specializing in Data Science

Department of Information Technology

Sri Lanka Institute of Software Engineering

October 2022

DECLARATION

I declare that this is my own work, and this Thesis 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, I hereby grant to Sri Lanka Institute of Information Technology, the non-exclusive right to reproduce and distribute my Thesis, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: ……. Date: 01/10/2021

The above candidate has carried out this research thesis for the Degree of Bachelor of Science (honors) Information Technology (Specializing in Information technology) under my supervision.

Signature of the supervisor: ……. Date: …….

ABSTRACT

Carbon dioxide has been identified as the primary cause of global climate change, which has gotten a lot of attention around the world. Among all greenhouse gases, carbon dioxide is the most prominent gas that causes carbon emissions in the environment. The entire amount of carbon dioxide (CO_2) released by human activity over time is referred to as the carbon footprint. Every activity a human takes releases carbon dioxide into the atmosphere. It is critical to understand how much carbon dioxide a person emits. Vehicle emissions, energy, firewood, air conditioning, refrigeration, and other variables can all contribute to carbon emissions. All these variables can be measured by the units in charge of them. The goal of this study is to verify each factor's measurement units, convert unverified units to the actual emission factor unit. Processing text data that is a mixture of natural language and formal languages, can be measurements or units, is required for a variety of natural language techniques. This study separates unit verification and conversion into two phases and presents a text classification technique that enables a model to perform unit classification to verify the given consumption unit with the emission factor unit as well as a unit conversion sector that allows for the transformation of different class unit into the perspective emission factor units using a unit conversion factor matrix in order to calculate carbon emissions.

KEYWORDS – Carbon footprint, Unit verification, Unit conversation, NLP, Carbon emissions calculation, Text classification, Unit conversion factor matrix

ACKNOWLEDGMENT

I want to sincerely thank everyone who assisted me in finishing my dissertation, especially my supervisors, Ms. Anjalie Gamage and Ms. Sanjeevi Chandrasiri, both of the faculty of computers at the Sri Lanka Institute of Information Technology. In addition, I want to express my gratitude to the panel's Prof. Koliya Pulasinghe, Mr. Vishan Danura Jayasinghearachchi, and Ms. Hasarangi Dhananjana Withanage, all of the Sri Lanka Institute of Information Technology's faculty of computers, for their insightful remarks. I also acknowledge the practical domain expertise that Dr. Daniel N. Subramaniam, Faculty of Engineering, University of Jaffna, provided as my external supervisor.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviation	Description
GHG	Greenhouse gas
CO2	Carbon dioxide
NLP	Natural language processing
DL	Deep Learning
SVM	Support Vector Machine

PCA	Principal Component Analysis
TLDDL	Transfer Learning Discriminative Dictionary Learning
NB	Naive Bayes
LR	Logistic Regression
IPCC	Intergovernmental Panel on Climate Change

1. INTRODUCTION

1.1. Carbon Footprint

Today's excessive carbon dioxide emissions have made global warming worse, posing a serious threat to the long-term viability of human society. The international community is now struggling with the issue of how to lower carbon dioxide emissions. The term "carbon footprint" refers to the entire amount of greenhouse gas (GHG) emissions that a person, company, event, or product produces, whether directly or indirectly [5]. The term "carbon emission" refers specifically to the total amount of carbon dioxide (CO2) emissions for which an individual or organization is responsible. Most human activities, including unsustainable production and consumption patterns, emit GHG either directly or indirectly. The most frequent Greenhouse Gas emitted by human activity is Carbon Dioxide (CO2), both in terms of quantity released and overall impact on global warming.

Emissions of carbon dioxide come from both natural and artificial sources. Decomposition, oceanic release, and respiration are examples of natural sources. Human-caused factors include the manufacture of cement, deforestation, and the use of fossil fuels like coal, oil, and natural gas [6]. Carbon dioxide levels in the atmosphere have increased significantly since the Industrial Revolution as a result of human activity, reaching risky levels that haven't been observed in the last 3 million years. A natural balance that has prevailed for thousands of years prior to human interference has been upset despite the fact that human-caused carbon dioxide emissions are significantly lower than natural emissions. This is due to the fact that natural sources and sinks remove roughly equal amounts of carbon dioxide from the atmosphere. Due to this, CO_2 levels were kept under control and below safe limits. The natural equilibrium has been upset by human-caused emissions, which have increased atmospheric carbon dioxide without decreasing it.

1.2. Carbon Dioxide emission: Human Activities

Human-related carbon dioxide emissions have grown since the Industrial Revolution. The environment frequently experiences a rise in carbon dioxide concentrations as a result of human activities like deforestation and the combustion of fossil fuels like oil, coal, and gas. Coal, natural gas, and oil burning account for the vast bulk of all human-produced carbon dioxide emissions.

Wooded area removal and other agricultural growth, as well as a few industrial operations such as cement manufacture, contribute to the relaxation.

1.2.1. Electricity

The economic sector that generates the highest man-made carbon dioxide emissions is the electricity and heating era. This company used fossil fuels to manufacture a small amount of carbon dioxide. This business operation is typically reliant on coal, which has the highest carbon-in-depth of all fossil fuels and hence has a substantial global carbon footprint. In practically all advanced nations, fossil fuel burning provides the majority of energy. Depending on the power blend of the local energy supplier, the energy consumed at home and at work is likely to have a significant impact on greenhouse gas emissions.

1.2.2. Transportation

Human carbon dioxide emissions are mostly emitted by the transportation industry. Transporting products and people throughout the world resulted in substantial carbon dioxide emissions from fossil fuels. Transportation-related emissions have increased fast during the 1990s, more than doubling in less than a decade. A sizable share of the sector's carbon dioxide emissions are brought on by the road transportation industry. The main sources of emissions in the transportation sector are cars, freight vehicles, and light-duty trucks, and emissions from all three have been rising steadily.

1.2.3. Industry

The third best supply of man-made carbon dioxide emissions is the commercial zone. Manufacturing, building, mining, and agriculture are all a part of the commercial zone. The fundamental enterprise is production, that's divided into five categories: paper, food, petroleum refineries, chemicals, and metal/mineral goods. Enormous volumes of every shape of greenhouse gas, however in particular big quantities of CO2, are produced with the aid of using production and business operations. This is because of the reality that many business centers use fossil fuels to generate the warmth and steam required at numerous levels of the producing process. For example, companies withinside the cement zone need to warmness limestone to 1450°C a good way to remodel it into cement, that's executed with the aid of using burning fossil fuels [8].

1.3. Units Classification Using NLP

Units can be varied based on the different carbon emission factors. This study focuses on unit verification, which is widely recognized as a difficult problem due to the vast range of carbon emission parameters. Text classification is the process of categorizing textual information into ordered groups [10]. It is also known as textual content tagging or textual content categorization. Text classifiers can be used to consistently identify unit classes from given texts using Natural Language Processing (NLP) and then assign a set of pre-defined tags or classes simply relying entirely on found measures [13].

1.4. Units Conversion Concept

Conversion of units is the process of converting between different units of measurement for the same amount, usually using multiplicative conversion factors. Units of measurement are required to quantify these values. There are instances when the measuring units employed do not correspond to the measurement choice and convenience, as well as the standards required for certain processes and applications. It is critical to convert such units to the point where they can be comprehended and applied correctly.

2. BACKGROUND & LITERATURE SURVEY

2.1. Background

Numerous NLP applications, such as question-answering, spam detection, sentiment analysis, news categorization, user intent classification, and content moderation, may make use of text classification [10]. Text classification options include manual annotation and automated labeling. As the volume of text data in industrial applications increases, automatic text classification is becoming more important. Since the identified issue of carbon emissions poses a risk to the survival of the planet, animals, people, and eventually existence as we know it, they should be taken seriously. The amount of carbon emissions trapped in our environment causes global warming, which causes weather change, with consequences such as melting of the polar ice caps, rising sea levels, disruption of animal natural habitats, extreme weather events, and many other negative side effects that are dangerous to the planet and to human and animal life.

To limit carbon emissions, first determine how much carbon is emitted into the environment on a daily basis by a person's actions. It's difficult to quantify a person's total carbon emissions for a day since there are so many various sources that might emit carbon dioxide into the atmosphere, and the measurement results collected from the user can be in different units. The units must be classified according to the identified carbon emission sources. To categorize the various units into their perspective categories, a text classification model may be trained and applied.

A text classification technique using natural language processing will be used to classify classes of the unit provided by the user and units in the selected emission factor. It will be verified if the classes are the same using the detected classed similarity of the units. If those classes differ, a conversion matrix for distinct unit classes will be used to identify the conversion factor for unit change, and values will be converted using the conversion factor before calculating emission.

2.2. Literature Survey

2.2.1. Research A: Natural Language Processing Techniques for Extracting and Categorizing Finding Measurements in Narrative Radiology Reports

In order to create a healthcare organization whose clinical procedure outcomes are repeatable and predictable, this study proposes quantitative outcome criteria. In imaging research, measurements are the most prevalent kind of quantitative parameter. To extract and categorize data from narrative radiology reports, they developed two NLP engines. Both engines fared well in a formal evaluation. The more knowledge-intensive parts of the engine have been made available to the public in two technical appendices. The study aids in the automated interpretation of radiological results and may be included into software for the processing of medical data.

2.2.2. Research B: Applications of Deep Learning in News Text Classification

The classification of news text in this study is done using a combination of deep learning (DL) techniques. This study addresses the aforesaid challenges by including the three improvements stated below. Making a two-level categorization model is the first step. While the second-level model is used to categorize them, the first-level model is utilized to identify news events. This paper proposes a discrete vector to express text feature information, taking into account the contribution of each word in classification and computing the probability variance of each word to obtain the contribution of each word in classification, in contrast to prior research that used a

word vector to express text feature information. Finally, in the suggested model, the text characteristics are expressed using the word vector and the dispersion vector. The dispersion vector is used to show the relationship between words and categories, whereas the word vector is used to capture the semantic information between words. According to the experimental comparison and analysis reported in this work, the first-level model's recognition rate is 99.5 percent, and the second-level model's accuracy rate is 94.82 percent, showing that the model has a significant capacity for news event identification and categorization. Text preprocessing makes use of the public stop word list, however since the news events stop word list is not produced, some feature information is filtered out. Later, a specific stop word list for news stories can be created.

2.2.3. Research C: Efficient English text classification using selected Machine Learning Techniques

The major goal of this project is to use Weka as an experimental tool for feature selection, performance assessment, and text categorization. To apply our method, you can use R, Tensor Flow, Python, or a Matlab simulation program. The best feature, such as frequency, the first letter of the paragraph, the question mark, and the full stop, were chosen at the beginning using pre-processing. We employ supervised machine learning to extract text features from texts written in the English language. We also evaluated other machine learning methods, including NB, SVM, and LR, to give a comparison study. On the datasets we examined, the simulations demonstrate that the SVM performs better than the other machine learning methods. The assessment and comparison made use of a few carefully chosen criteria, including accuracy, recall, and F1 value. Finally, they talk about how well the machine learning techniques we chose performed. Our explanation makes it evident that each machine learning classification method has benefits and drawbacks that depend on the size of the datasets.

2.2.4. Research D: Sentiment Classification of News Text Data Using Intelligent Model

This article presented a transfer learning classification method for cross-domain text sentiment categorization. Based on the advantages of dictionary learning in knowledge reconstruction and sparse representation, they proposed integrating subspace projection and transfer learning into the dictionary learning framework. They employ the PCA term in the objective function to keep the discrimination knowledge while defining the following discrimination information preserved term

in the objective function, accounting for the within-class minimizing and between-class maximizing of sparse coding coefficients. Such an algorithm produces a domain-invariant dictionary to establish relationships between several domains. The results of the experiments show that the TLDDL algorithm has a high classification performance.

2.2.5. Research E: Deep Learning Based Text Classification: A Comprehensive Review

Deep learning-based models have outperformed conventional machine learning-based methods in a range of text classification tasks, including sentiment analysis, news categorization, question answering, and natural language inference. In this paper, they examine the technical contributions, commonalities, and advantages of more than 150 deep learning-based text classification models developed in recent years. Additionally, they provide a list of more than 40 widely used datasets for text categorization. Finally, they discuss potential directions for future study and offer a quantitative analysis of how various deep learning models performed on well-known benchmarks.

3. RESEARCH GAP

As per the literature review, there are several researchers have been done on the domain of text classification using NLP. Previously text classification concept was applied on different kind of topics such as movie reviews, question and answering, news text classification, spam mail detection, topic labeling, tagging content, etc. The majority of these applications have to take the unstructured text data to process, classifying that kind of text is very extremely useful for variety of such purposes. Based on the past research analysis, there were no efficient algorithms or methods developed on classifying mathematical units of measurements for any of calculation purposes. Some of the scientific report analysis researchers have been focused on categorizing measurement from the documents. Class comparison measures, also known as "comparison measurements," generally indicate the dimensions of a current finding but are compared qualitatively to the dimensions of the same finding on a previous. When the dimensions of a discovery on current and past are equal, it may be more artistically preferable to use a comparative measurement and other calculations[1]. The past researchers have been used to approach the text classification in three different ways as rule-based methods, machine learning based methods, hybrid methods. The rule-based methods employ handmade linguistic rules. Making a list of terms linked to a specific column and then judging the text based on the occurrences of these words is

one technique to organize content. Words like "fur," "feathers," "claws," and "scales," for example, might aid a zoologist in locating animal-related writings on the internet. These methods need a great deal of subject expertise, take a long time to compile, and are difficult to scale.

To anticipate new text categories, a machine learning approach is used to train models on massive amounts of text data. The process of transforming text input into numerical data in order to train models is known as feature extraction. Two popular feature extraction methods are bag of words and n-grams. We may use a number of machine learning approaches to categorize text. The machine learning based approaches have been categories under naive bayes classifiers, support vector machine, deep learning algorithms. By comparing with traditional machine learning based techniques, deep learning models can perform well [10]. Hybrid methods combine the two techniques mentioned above. They create a classifier that can be fine-tuned in particular instances using both rule-based and machine learning approaches.

Table 3.1:Comparison of Unit Classification with Former Researches of Text Classification

Related-Works	**Classification of mathematical units/measurements**	**Deep learning based approach**	**Compression of units/ measurements**	**Convert values**
Research A	✓	✗	✓	✗
Research B	✗	✓	✗	✗
Research C	✗	✓	✗	✗
Research D	✗	✗	✗	✗
Proposed System	✓	✓	✓	✓

The aforementioned characteristics were identified as the primary research needs in previous studies and evaluations of text categorization algorithms utilizing NLP. The suggested system will fill the research gaps that have been discovered. The use of deep learning methodologies to classify mathematical units will aid in the development of new NLP concepts and techniques in the text classification arena.

3.1. Research Problem

Carbon emission is the known problem that cause the global warming resulting climate changes[7]. Every human being should know their cumulative amount of carbon that can be emitted to the environment by his/her own day to day activities.

This is probably because of a loss of studies approximately the significance of carbon footprint measuring in the company. Every year, carbon emissions must be measured in equal kilograms of CO2 (kg CO2 e) or equal heaps of CO2 to estimate how tons carbon every character emits (ton CO2 e). This calculation will yield an annual carbon footprint figure, permitting for a higher information of carbon footprint and the method of mitigation measures[11]. "kgCO2e/(m2-a)" is the functional unit, which represents the CO2 equivalent per square meter of building area per year. This unit of computation efficiently eliminates the impact of variable building sizes and design years, ensuring that accounting results are uniform and comparable[12].

Each year, a typical passenger car emits around 4.6 metric tons of CO2. This means that the typical gasoline automobile today has a fuel economy of gallons per gallon. When a gallon of gasoline is burned, roughly 8,887 grams of CO2 are released[8][9].

The emission factor approach, commonly known as the IPCC inventory method or the emission factor method, is a popular method for assessing carbon emissions today. Carbon emission factors must be determined before carbon emission estimates for buildings can be performed[12].

The environment may be used to identify numerous carbon emission variables. The many factors may make it difficult to determine the right emission factor unit. Because a person's input may be incorrect, the overall quantity of carbon emissions may be incorrectly calculated. When utilizing a voice discussion to collect emission information, each informant has their unique manner of describing the emission unit. In terms of unit measurements, emission records come in a variety of forms. As a result, before computing a person's total carbon emissions for a day, the user's stated figures should be checked.

The validation cannot be done manually; instead, it must be verified by some form of intelligence support system. Users can select emission factors with different units, resulting in the given carbon emission unit and the values do not match the actual carbon emission factor unit, and users can

select emission factors with different units, resulting in the given carbon emission unit and the values do not match the actual carbon emission factor unit. The provided unit may be converted to compute the carbon emission by determining the conversion factor.

4. OBJECTIVES

4.1. Main Objectives

The main objective of this research component is to design and implement a unit verificarion and conversion component to classify different units, convert them to same format. When computing carbon emissions, keeping track of units is critical. Emission factor standards provide values in a variety of units. It's crucial to make sure the emission factor and activity units are the same. This is a time-consuming task for the person in charge. As a result, the goal of my study is to determine the rate of carbon emission in a common unit that can be used to compute the rate of carbon emitted by a given element.

4.2. Specific Objective

- Research the ways to create unit classification using natural language processing.
- Collect carbon emission units and measurements related data set.
- Identify the real units of emission factors gathered in the data set.
- Preprocess collected text dataset to train the model.
- Choose the best algorithm to classify different units.
- Implement unit conversion factor matrix to identify the conversion factor to convert the different units of measurements.

5. METHODOLOGY

5.1. System Overview

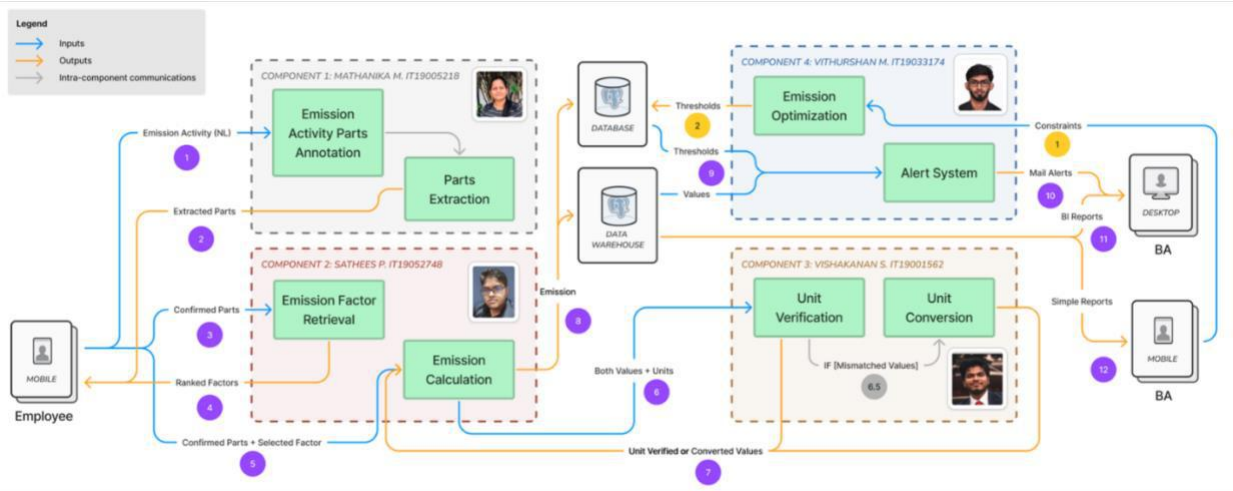

Figure 5.1: Overall system architecture

In above figure, it illustrates the high-level system overview diagram of the Carbon emission calculation system which is proposed. There are mainly two ways a user can input the carbon emission detail for a day such as question and answering as a text or voice input. Natural language processing techniques will be used to extract necessary bits from the audio recording. Text is generated from the speech input. To extract carbon emission information, the text data is preprocessed, converted, and cleaned. The voice assistant will ask the user to explain if there is any uncertainty or missing information. It will also receive confirmation on emission parameters that have been identified as important. A factor finding module will return appropriate factors to the voice assistant for confirmation based on the emission activity sections detected by the voice assistant. Due to the technical nature of the emission factors, a natural language-based information retrieval system will be employed to quickly locate important emission factors.

The given emission detail is subjected to unit validation and conversion in order to categorize the units of measurement and match the emission factor data provided by the user. A text classification strategy employing natural language processing will be used to classify classes of the unit given by the user and units in the specified emission factor. It will be checked if the classes are the same using the detected classed similarity of the units. If those classes differ, a conversion matrix for

distinct unit classes will be used to calculate the conversion factor for unit change, and values will be converted using the conversion factor before computing emission.

The emission optimization sector gives the alerts to the user based on the maximum emission value provided by the solution, maximum thresholds are exceeded. These identified subsystems are:

1. Efficient emission factor searching.
2. Real-time collection of emission data.
3. Unit checking and conversion to match the emission factor's unit.
4. Emission optimization for the emission source constraints.

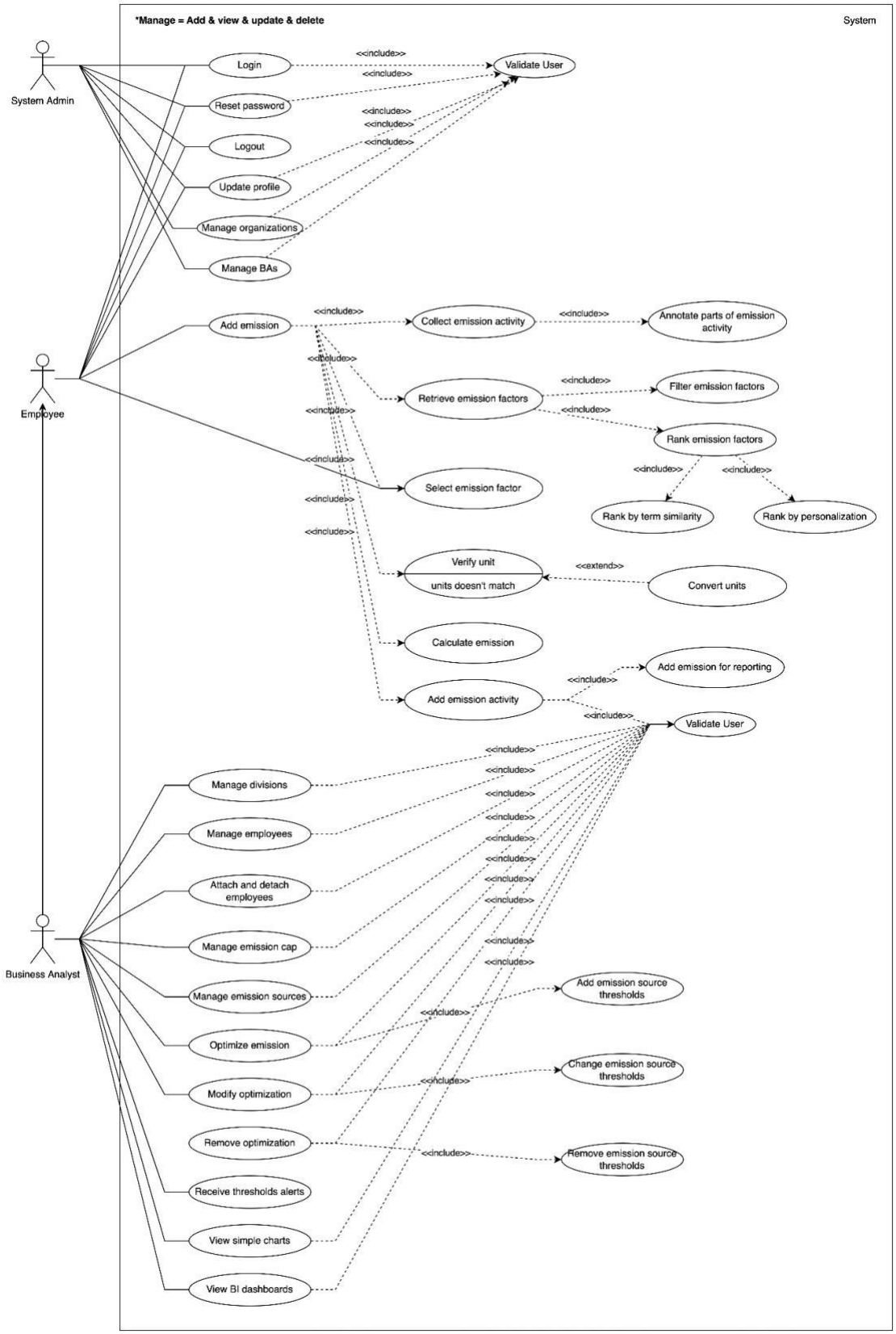

Figure 5.2:system use case diagram

5.2. Component Overview

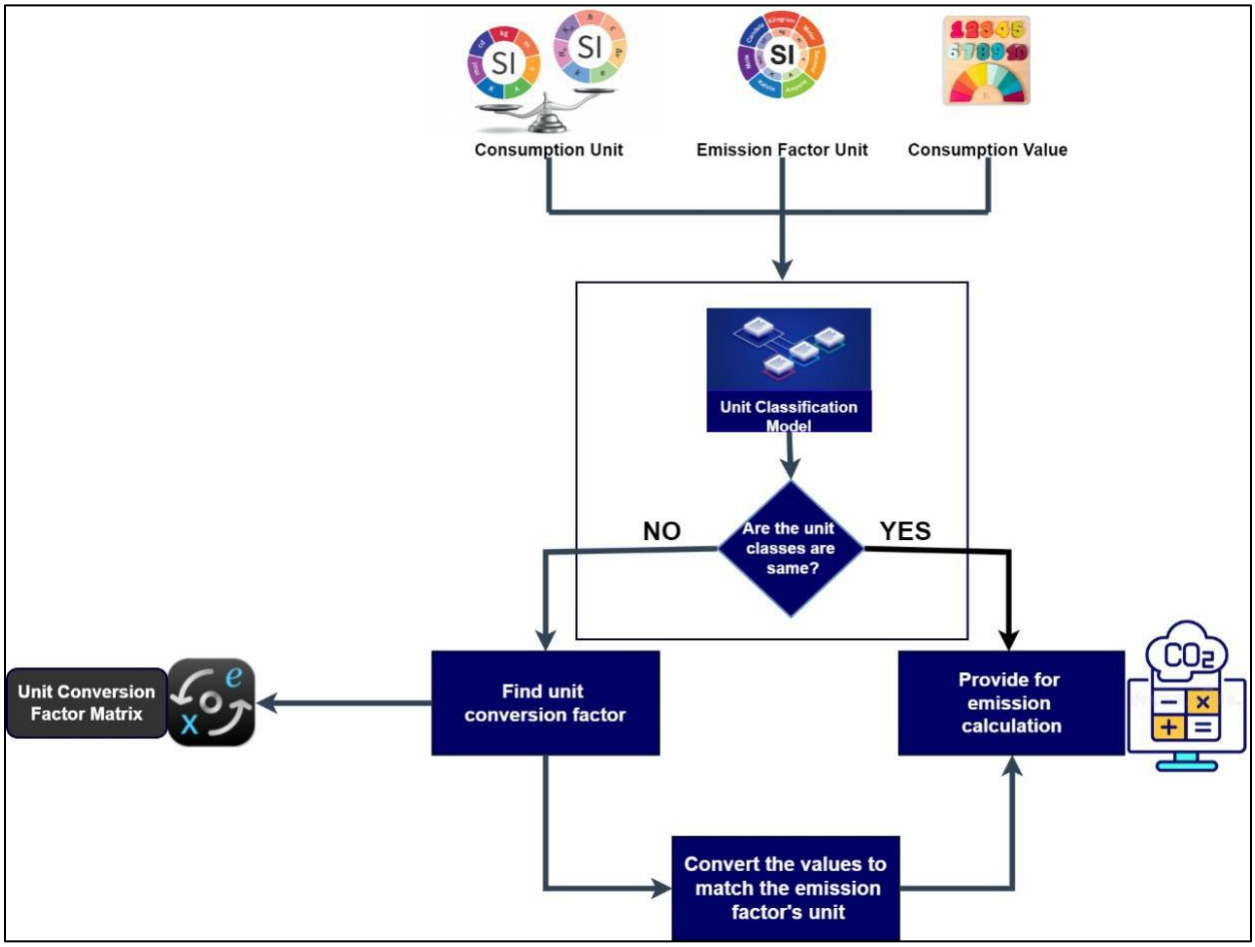

Figure 5.3: High level Diagram of Unit Classification and Conversion Sub System

The Individual research component which is conducted by myself includes the unit verification and conversion subsystem where the units can be verified along with the provided factor of carbon emission. The high-level architecture of the subsystem is shown in Figure 6.2. The measurements and the units of the carbon consumption is given to the subsystem with the emission factor unit which has been identified by the system. Unit classification model is used to verify if the unit classes are same or not. The consumption unit and the emission factor unit is different the unit conversion factor need to be identified from the unit conversion factor matrix and the different measurement values will be converted to the same emission factor unit. After finish the unit classification and conversion process, the output will be given for the carbon emission calculation.

The goal of this stage is to classify the different units and convert different units to the same format to calculate the carbon emission efficiently.

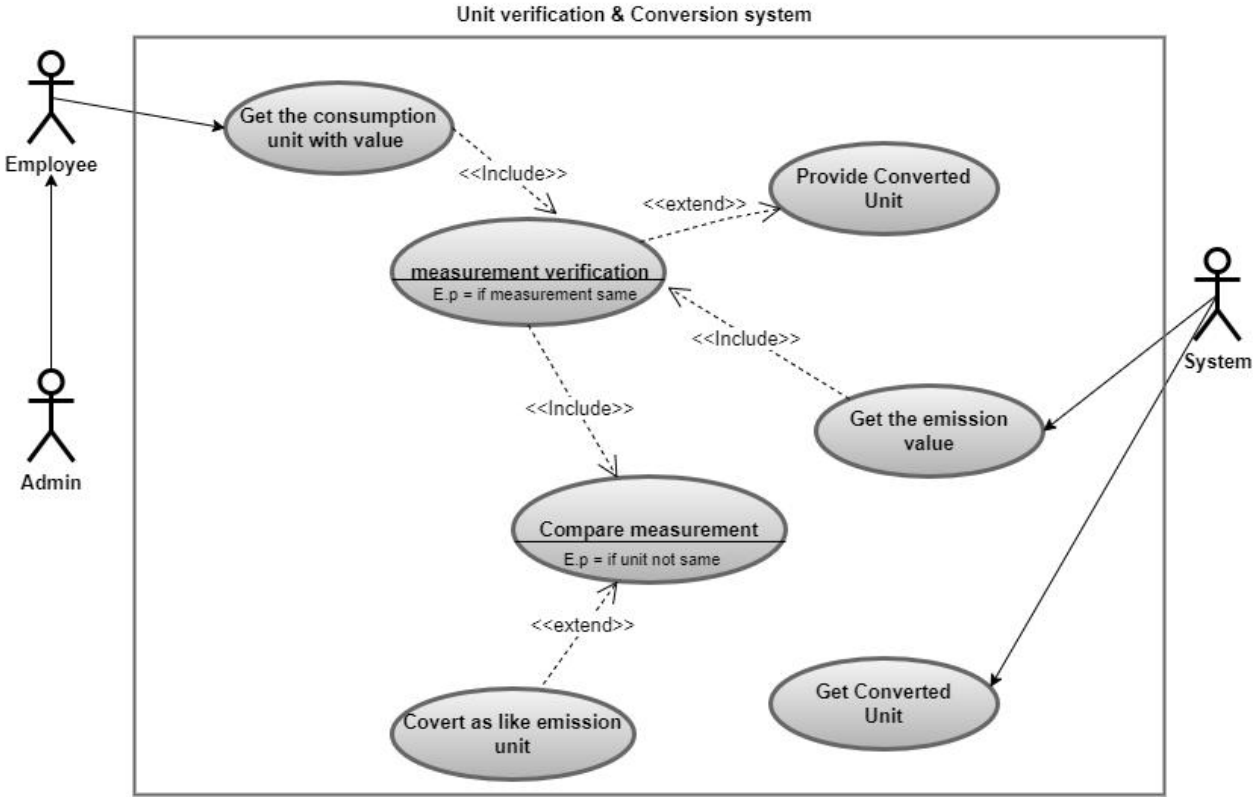

Figure 5.4: unit verification & conversion Use case diagram

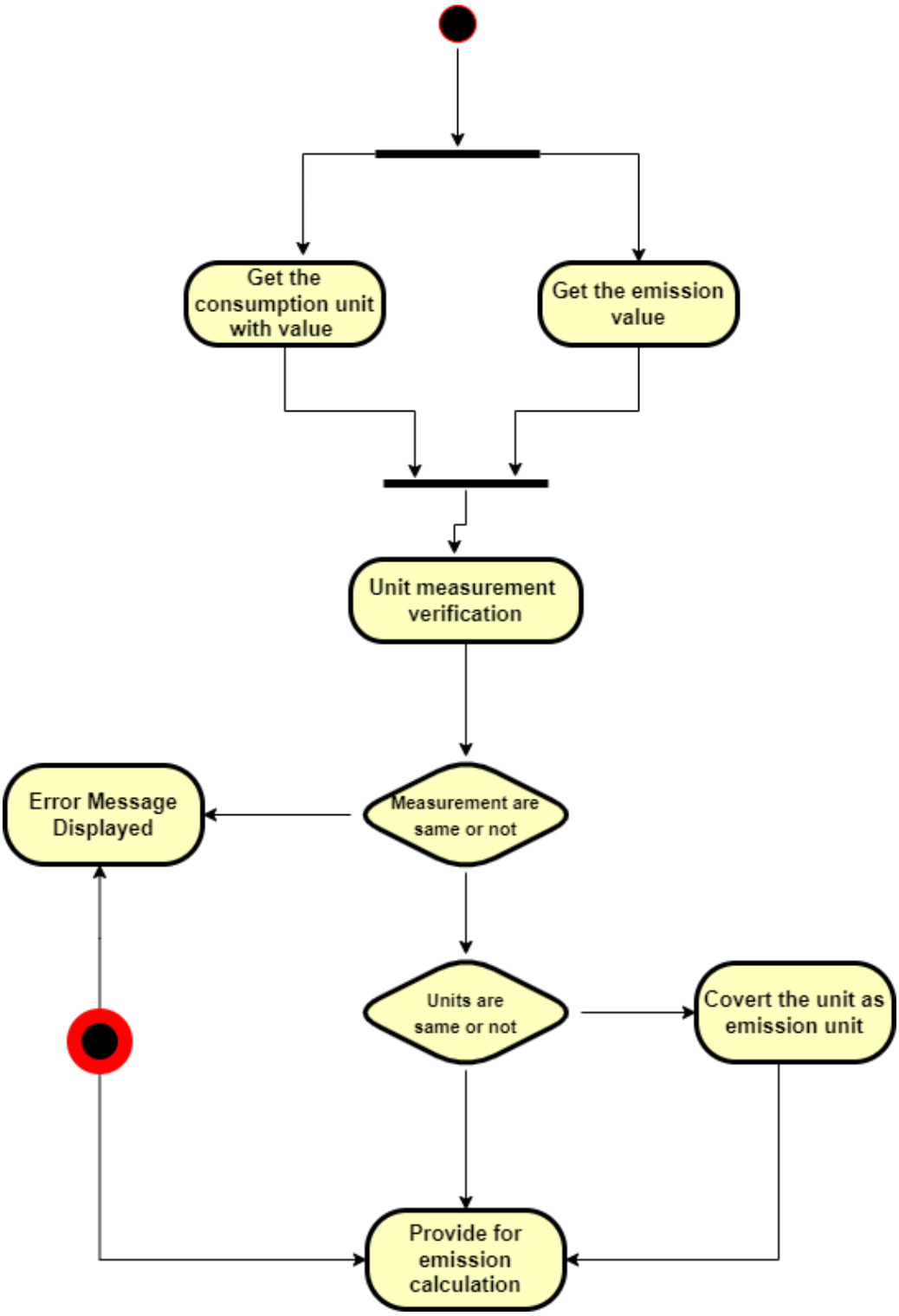

Figure 5.5:unit verification & conversion Activity diagram

5.3. Preliminaries

5.3.1. Text classification

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.

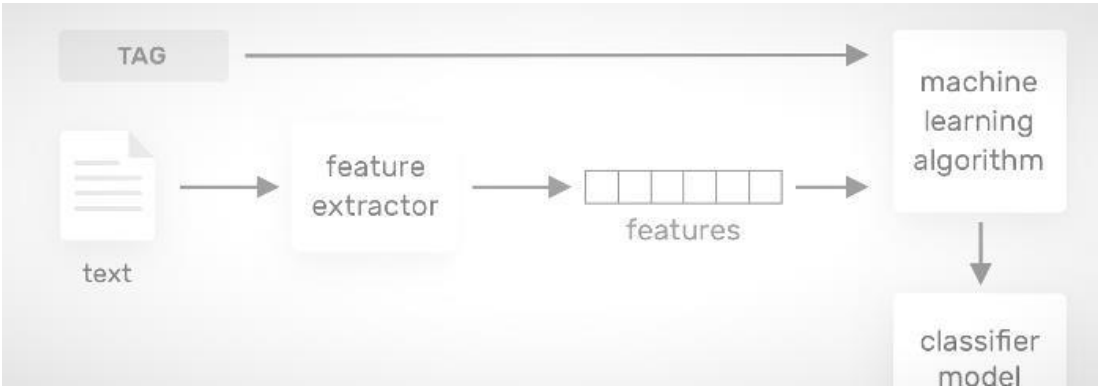

Figure 5.6:train the text data set

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.

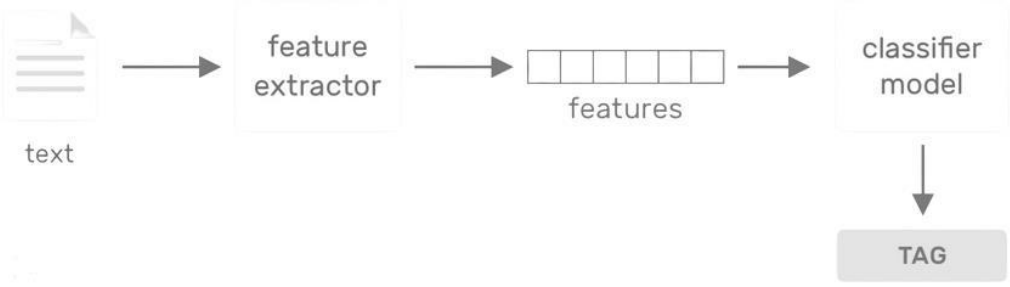

Figure 5.7:predict through the trained the text data

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.

5.3.2. 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.

5.4. Technologies and 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:

$$\frac{\boldsymbol{x}^\top \boldsymbol{y}}{|\boldsymbol{x}||\boldsymbol{y}|} \in [-1, 1].$$

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.

Several hugging face deep learning models are being employed at this point for training. My main emphasis is on the BERT, Distil-Bert, ROBERTA, and GPT-2 models.

5.4.1. BERT

BERT and other Transformer encoder designs have excelled in a wide range of NLP applications (natural language processing). They provide natural language vector-space representations appropriate for deep learning algorithms. The Bidirectional Encoder Representations from Transformers (BERT) family of models leverages the Transformer encoder architecture to interpret each token of input text in the context of all tokens that came before and after it. BERT models are frequently trained on a big corpus of text before being tailored for certain applications.

5.4.2. Distil-BERT

The transformers model Distil-BERT was pretrained on the same corpus using the BERT base model as a teacher. It is smaller and faster than BERT. This means that the BERT base model was automatically used to automatically generate inputs and labels from those texts during pretrained using only the raw texts, with no human labeling of them in any way (which explains why it can use a significant amount of data that is easily accessible to the public).

5.4.3. ROBERTA base model

A huge English data corpus was used for the self-supervised pretrained transformers model named ROBERTA. This proves that a machine-learning algorithm was employed to generate inputs and labels from those texts after it had been trained on simply the raw texts without any human labeling (which explains why it may use a ton of data that is readily available to the public).

For the purpose of Masked Language Modeling (MLM), it was particularly pretrained. The model must predict the hidden words once a phrase is presented and 15% of the input words are chosen at random to be hidden. This is in contrast to autoregressive models like GPT, which internally conceal the following tokens, and standard recurrent neural networks (RNNs), which normally perceive the words in order. As a result, the model can discover a two-way representation of the text.

If you have a dataset of labeled sentences, for instance, a standard classifier can be trained using the features generated by the BERT model as inputs. By doing this, the model develops an internal representation of the English language from which elements helpful for later tasks can be extracted.

5.4.4. GPT-2

A substantial English data corpus was used for the self-supervised pretrained transformers model GPT-2. This proves that a machine-learning algorithm was employed to generate inputs and labels from those texts after it had been trained on simply the raw texts without any human labeling (which explains why it may use a ton of data that is readily available to the public). The following word was explicitly taught to be implied in sentences.

5.5. Implementation

I trained among those models and got the best one for my unit verification. So, through the unit verification, I can find the suitable measurement for that employee's entered emission unit.

The implementation process of Unit Verification Model is divided into 5 main subtasks, which are,

1. Data set collection
2. Preprocess Phase
3. Feature Extraction Phase
4. Training Phase
5. Testing Phase
6. Evaluation Phase

5.5.1. Data set collection

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.

5.5.2. Preprocess Phase

I see that text data isn't always tidy in my circumstances. One of the most crucial processes in the classification pipeline is text preprocessing because data from various sources have distinct characteristics. So I did several steps preprocessing purpose.

- Eliminating Punctuation and Cleaning Special Characters

  The word2vec embeddings I'm going to employ for our classification job heavily influence my preprocessing pipeline. My preprocessing ought should, in theory, be identical to the preprocessing carried out before to word embedding training.
- Cleaning Quantities
- Eliminating Misspellings

  Finding typos in the data is always helpful. In order to improve embedding coverage, I replaced terms with their correct spellings because such word embeddings are not available in the word2vec.
- Sequence creation representation

  I don't actually have to hand-engineer features from the text input, which is one of the factors that has made deep learning the "go to" option for NLP. In order to learn the structure of text, the deep learning algorithms use a sequence of text as their input, just like a human would. Machines expect their data in numerical form because they cannot interpret speech. In order to represent our text data, I would like to use a series of numbers. that's why I did this section.

5.5.3. Feature Extraction Phase

The feature extraction phase is a very critical part of creating a model and training it. After this phase data will be ready to train a model, therefore this step is needed to be done very carefully because it will affect the accuracy of the model. Words in the text dataset consist of discrete and categorical features which must be mapped to real-valued vectors in order to be used by the algorithms. The given emission factors of the data set needed to be find the correct emission factor units. After find the real units, several techniques can be used to convert a text dataset into a vector. The vector will be consist of the emission factors and the measurements, given emission factor units , real emission factor units to train the model.

Techniques for feature extraction are required In order to create results for the test data, machine learning algorithms learn from a predefined set of features from the training data. However, the primary issue with language processing is that machine learning techniques cannot be used to

directly handle raw text. In order to turn text into a matrix (or vector) of features, I used certain feature extraction algorithms. I employed two feature extraction techniques, namely:

- Bag-of-Words

Bag-of- The usage of words is one of the most fundamental procedures for transforming tokens into a group of features. The classifier in the BoW model, which is used to categorize documents, is trained using each word as a feature.

- TF-IDF

The acronym TF-IDF stands for term frequency-inverse document frequency. It calls attention to a specific issue that might not surface frequently in our corpus but is nonetheless very important. When a word is used frequently in a document, the TF-IFD score increases, and when it is used less frequently in a corpus of documents, the score decreases.

- Word Embeddings

Word embedding is a technique for encoding texts and documents using a dense vector representation. The context of a word in text, which is based on the words that appear around it, determines where that word is located within the vector space. Word embeddings can either be taught using the input corpus itself or pre-trained word embeddings like Glove, FastText, and Word2Vec. In the four phases that follow, I applied word embeddings that had already been learned to the model.

1. The pretrained word embeddings are loaded.
2. I a tokenizer object was created
3. Text documents were converted into token sequences and padding.
4. Built a mapping between tokens and their associated embeddings

5.5.4. Model implementation phase

The model was used to identify and categorize the various carbon emission units in the provided dataset. The various units were submitted together with the various measures' converted values to find the conversion factor from the unit conversion factor matrix.

Figure 5.8:model configuration for training

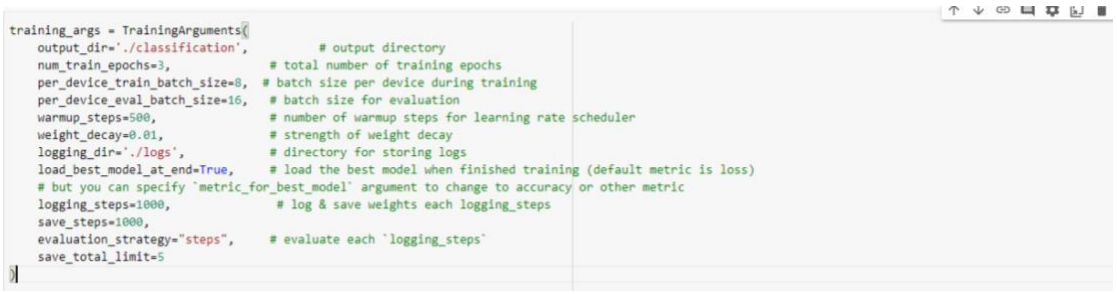

Figure 5.9:training arguments setting for training phase

Here I configured the model for the training phase, so first of all I defined the model and configured it, and after that I set training arguments with 16 epochs.

5.5.5. Training Phase

How well the model categorizes units in the end depends on how well it performs throughout training. When training the model, both the quality of the training data and the method utilized must be taken into account. Training and validation and testing data are the two categories into which training data is separated. Considerations include algorithm-model complexity, performance, interpretability, resource requirements for the computer, and speed. Algorithm selection may be time-consuming and challenging because so many objectives must be balanced.

```
[ ] #CUDA_LAUNCH_BLOCKING = "1"

    trainer = Trainer(
        model=model,                         # the instantiated Transformers model to be trained
        args=training_args,                  # training arguments, defined above
        train_dataset=train_dataset,         # training dataset
        eval_dataset=valid_dataset,          # evaluation dataset
        compute_metrics=compute_metrics,     # the callback that computes metrics of interest
        tokenizer = tokenizer
    )

    # train the model
    trainer.train()

/usr/local/lib/python3.7/dist-packages/transformers/optimization.py:309: FutureWarning: This implementation of AdamW is deprecated and will be removed in a futu
  FutureWarning,
***** Running training *****
  Num examples = 15076
  Num Epochs = 3
  Instantaneous batch size per device = 8
  Total train batch size (w. parallel, distributed & accumulation) = 8
  Gradient Accumulation steps = 1
  Total optimization steps = 5655
[5655/5655 1:28:00, Epoch 3/3]
```

Figure 5.10:training the deep learning model

5.5.6. Testing Phase

After the model has been trained, it must be tested using data that has never been seen before. The test results must be assessed against the performance evaluation criteria. To prevent the problem of unstable data, repeat this step by altering the training and test data during the training and testing procedure.

5.5.7. Evaluation Phase

The evaluation process starts after the first four steps are finished. The main objective of this stage is to assess the precision and effectiveness of the trained models in categorizing various input units. To identify which parts of the system need to be increased and improved, I want to compare the model's loss and accuracy, as well as gather various amounts of data. I want to survey students who are selected at random after the integration of the full product.

```
[ ] # evaluate the current model after training
    trainer.evaluate()

***** Running Evaluation *****
  Num examples = 3770
  Batch size = 16
[236/236 02:20]
tensor([[6],
        [0],
        [6],
        ...,
        [6],
        [6],
        [1]])
/usr/local/lib/python3.7/dist-packages/seqeval/metrics/sequence_labeling.py:171: UserWarning
  warnings.warn('{} seems not to be NE tag.'.format(chunk))
/usr/local/lib/python3.7/dist-packages/seqeval/metrics/sequence_labeling.py:171: UserWarning
  warnings.warn('{} seems not to be NE tag.'.format(chunk))
/usr/local/lib/python3.7/dist-packages/seqeval/metrics/sequence_labeling.py:171: UserWarning
  warnings.warn('{} seems not to be NE tag.'.format(chunk))
/usr/local/lib/python3.7/dist-packages/seqeval/metrics/sequence_labeling.py:171: UserWarning
  warnings.warn('{} seems not to be NE tag.'.format(chunk))
```

Figure 5.11: evaluate the trained model

	precision	recall	f1-score	support
0	0.64	0.73	0.68	319
1	0.45	0.83	0.58	389
2	0.81	0.64	0.71	394
3	0.64	0.57	0.61	392
4	0.55	0.78	0.64	385
5	0.77	0.52	0.62	395
6	0.84	0.77	0.80	390
7	0.87	0.79	0.83	396
8	0.85	0.90	0.87	398
9	0.98	0.84	0.90	397
10	0.93	0.96	0.95	399
11	0.92	0.79	0.85	396
12	0.59	0.53	0.56	393
13	0.82	0.82	0.82	396
14	0.84	0.84	0.84	394
15	0.83	0.89	0.86	398
16	0.68	0.86	0.76	364
17	0.97	0.86	0.91	376
18	0.66	0.50	0.57	310
19	0.53	0.31	0.40	251
avg / total	0.77	0.75	0.75	7532

Figure 5.12: Model accuracy evaluation

5.6. Commercialization

Power sector commercialization is projected to have a variety of favorable environmental effects. First, by commercializing the concept of cost recovery, greater incentives might be created for more effective management and operation of the available capacity as well as more effective transmission and distribution. Cost recovery will also encourage utilities to implement end-use efficiency initiatives, particularly to cut losses in rural areas where the income earned frequently falls short of the marginal cost of energy provision. The process of commercialization typically involves the following steps:

1. Research

The moment a research finding is made, the commercialization process starts. In our scenario, our team has already completed our study concepts.

2. Disclosure of an invention

The first step in formally documenting our idea is to send a disclosure of invention to TCS. This is a confidential document that describes your innovation in great detail so that we can assess our commercialization alternatives. In our situation, the documentation for our invention idea is now complete.

3. Marketing

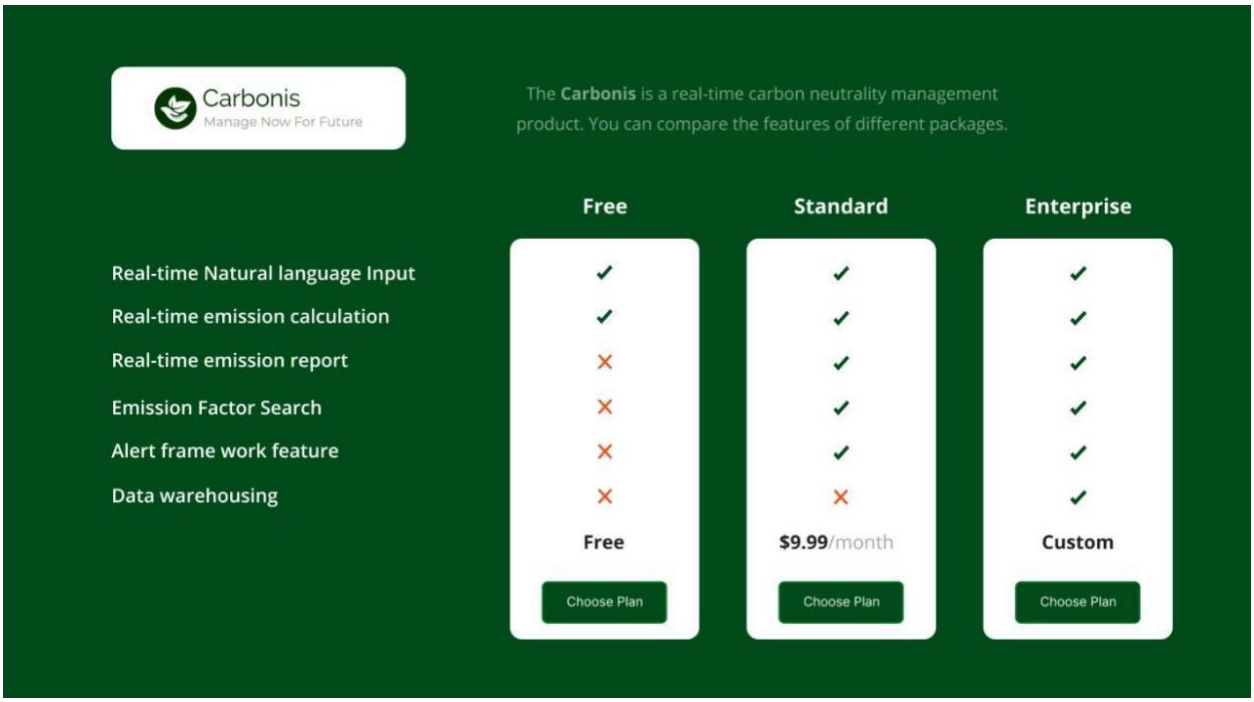

Figure 5.13: Pricing Plan

In order to commercialize the technology, we aim to license the invention to an already-existing business or support the establishment of a new one. We are now looking for business partners. We intend to offer some pricing packages, such as Free, Standard, and Enterprise, in order to impress our marketing partners.

4. Licensing

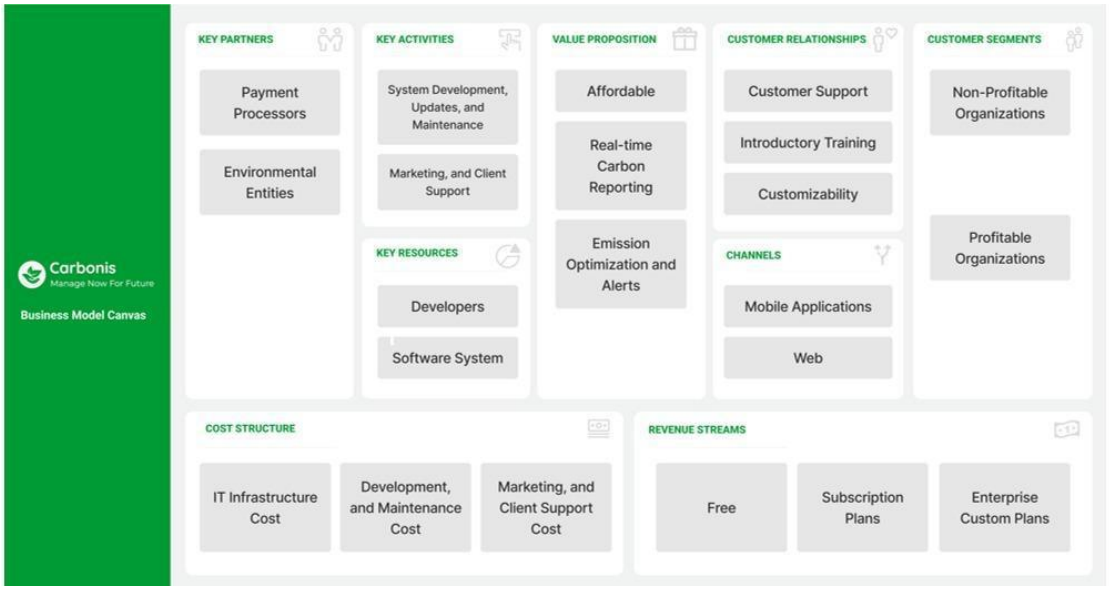

Figure 5.14:Business model canvas

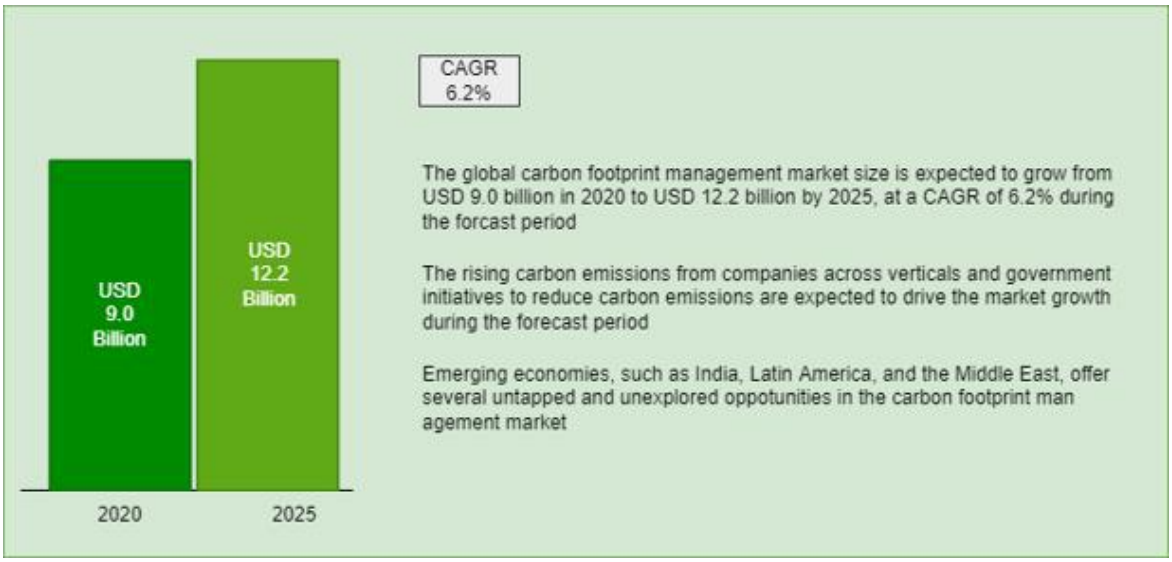

Figure 5.15: Economical range for the global emission problem.

Based on its capacity to commercialize the technology for the benefit of the broader public, a licensee is picked. The greatest option is occasionally a well-established business with expertise

in related sectors and technologies. Other times, a new company's intensity and focus are preferable. Therefore, we are also creating a business model for that company idea.

According to economic analysis, market emissions are a significant economic concern in the actual world, hence our product is viable on the market.

5. Investment

Figure 5.16: Product promotion pamphlet

The income from investment licensing that UConn receives is given to inventors and their academic departments. Additional research, teaching, and engagement in the technological commercialization process are supported by these earnings. Therefore, we have finally decided to launch the app in the real world for business purposes.

6. RESULTS & DISCUSSIONS

In the results portion of this chapter, experiment results are presented, and the research findings section contains the conclusions of those studies. The conclusion and the rationale for these conclusions are summarized in the discussion section.

6.1. Results

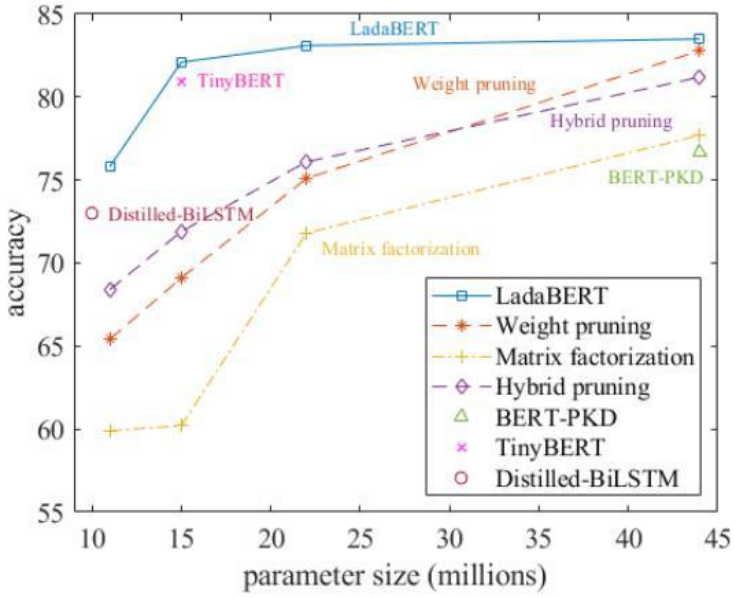

Figure 6.1: Model Comparison

The example shows how language models based on transformers predominate in all TC tasks. BERT and Lada-Bert produce cutting-edge outcomes on the bulk of datasets. Furthermore, the use of graph convolutional networks in conjunction with TC for document representation demonstrates success in the extraction of useful features, particularly when node embeddings are initiated with contextual representations generated by BERT-like models (Bert, Roberta). The model architecture is rather simple compared to BERT and other deeper language models, but it nevertheless achieves the maximum accuracy on a binary SA dataset. The authors contend that for this kind of assignment, tailored document representations are more important than classifier choice. This model stresses the importance of the attention process and uses recurrent blocks (GRU) to enhance seq2seq architectures.

Figure 6.2: result of unit verification

Given units are categorized as measurements in this instance. Then, all values and measurements are separated and shown as though they were a dictionary.

The use of broad multi-task benchmarks is the newest development in the assessment of deep language models. The findings are shown in the above graph. These benchmarks, which are not tailored specifically for TC, assess how well a model can generalize to different tasks. The provided score represents the model's performance characteristics averaged over all tasks.

My unit converter part is shown in the figure above. Then, units are checked; if necessary, they must be converted to emission factor units if they are not the same. This implies that any value entered with any unit will be transformed to the appropriate format.

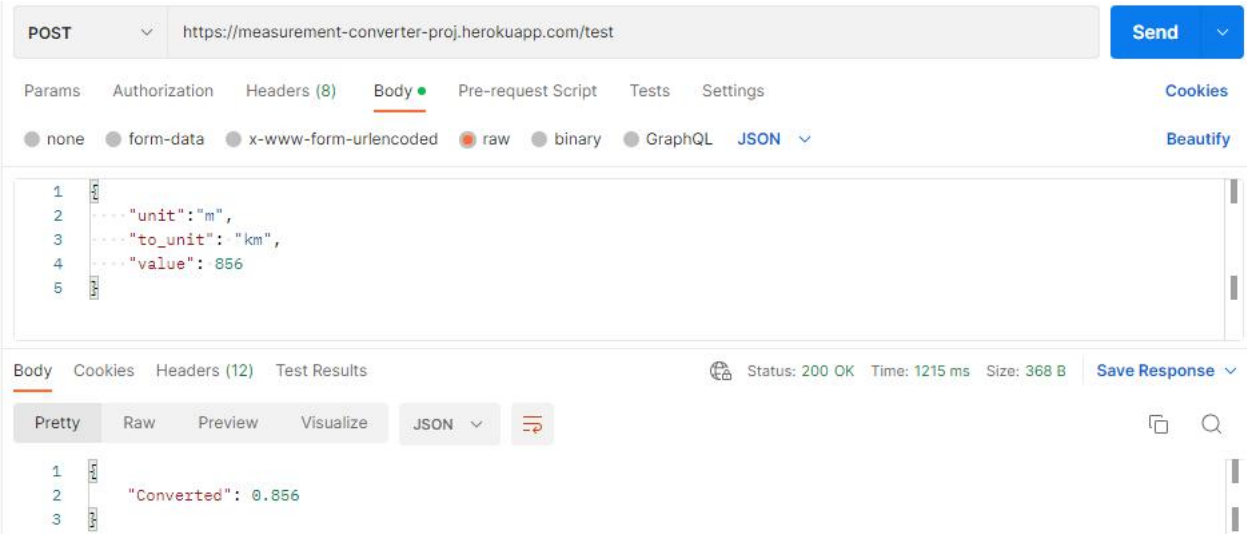

Figure 6.3: unit conversion API results

6.2. Research 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.

6.3. Discussion

The basic concept of my system is unit verification and conversion. Even without unit verification and conversion, emission calculation is possible. To do so, user inputs must be limited to the shortlisted possible units so that emission calculations can be performed. But that method will affect user satisfaction. It is not a good approach to assume that users must be familiar with those restricted units which are required by the system. So, in order to get the user's satisfaction, the application should verify whatever the input user enters. To verify the user input, a deep learning model will be applied with 80 % accuracy. Once the verification is done, the system should

compare the user input unit with the emission factor unit. If the comparison is positive, an input unit will be provided for emission calculation if its negative application converts the input and handovers it for conversion with the use of the regax algorithm. So the system's functionalities are totally based on the user input.

7. RESOURCES USED

- React Native: React Native is a JavaScript framework that makes it possible to create mobile apps much more quickly than through native programming, and it supports both iOS and Android natively. React.js, a component-based framework for developing front-end web apps and user interfaces, was used to build it.
- PyCharm: For usage with the Python programming language, PyCharm is an Integrated Development Environment (IDE). It was made by the Czech company JetBrains. It provides code analysis, a graphical debugger, an integrated unit tester, and integration with version control systems in addition to supporting Anaconda (VCSes).
- Django: A sophisticated Python web framework Django encourages rapid development and efficient, useful design. It was made by skilled programmers and takes care of a lot of the hassles related to web development, freeing you up to focus on creating your app without having to reinvent the wheel. It is both free and open source.
- Postman: Postman is an API platform for building and using APIs. Postman enhances cooperation and streamlines each step of the API lifecycle in order to assist you in designing better APIs more quickly.
- GitLab: GitLab is a free online Git repository that offers wikis, issue tracking, open and private repositories, and Git. With the help of this comprehensive DevOps platform, developers can control every aspect of a project, including planning, maintaining code sources, monitoring, and security.
- Colab: Google Research's Collaboratory, also referred to as "Colab," is a product. Data analysis, teaching, and machine learning are three areas where Colab excels. Through the browser, anyone may write and run arbitrary Python code.

8. Testing

Application development includes a challenging and essential step called application testing. Performance, usability, security, nonfunctional and functional testing tools, various platforms, and browsers are the main elements of an application testing. Moreover, it is crucial to guarantee the product's quality. Design testing, unit testing, module testing, integration testing, and acceptance testing are just a few of the stages that make up this process.

- Design Testing

Applications are tested in this step to make sure they have the functionalities needed. Testing the responsiveness of the interface and how the app should react to touch controls. Additionally, each UI designer was put through a test to determine whether they improved user experience.

- Unit Testing

Each module or function is tested individually during the unit testing process after implementation. everything is individually tested to make sure each function is operating properly. Every each member does this.

- Module Testing

Within a program, specific subprograms, subroutines, classes, or processes are tested as modules. This is carried out by a group member who did not implement the specific module.

- System testing

To guarantee that the entire system is operating properly, it will be tested. Anyone in the group is able to carry out this.

- Acceptance testing

The system is evaluated in this step for acceptability. This testing's objective is to compare the system's functionality to its commercial value.

Following are some test cases that we have used to test this unit component.

Table 7.1:Test case for check unit conversion.

Test Case ID	011
Test Case scenario	Check recommended output
Test Step	1.user should login the system 2.user enter consumption units same as emission factor unit 3.Calculate Emission
Test Data	Today travel distance is 7km
Expected result	No need convert
Actual result	Didn’t convert, in calculation system use 7km
Pass/Fail	Pass

Table 7.2: Test case for check unit conversion

Test Case ID	012
Test Case scenario	Check recommended output
Test Step	1.User should login the system 2.User enter consumption units different as emission factor unit 3.Calculate Emission
Test Data	Today travel distance is 7000m
Expected result	System need to convert as 7Km
Actual result	converted as 7Km
Pass/Fail	Pass

9. CONCLUSION

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.

10.REFERENCES

[1] Sevenster M, Buurman J, Liu P, Peters JF, Chang PJ. Natural Language Processing Techniques for Extracting and Categorizing Finding Measurements in Narrative Radiology Reports. Appl Clin Inform. 2015;6(3):600-110. Published 2015 Sep 30.

[2] S.L., Berrahou & Buche, Patrice & Dibie-Barthélemy, Juliette & Roche, Mathieu. (2013). How to extract unit of measure in scientific documents?

[3] Hundman, K., & Mattmann, C. A. (2017). Measurement context extraction from text: Discovering opportunities and gaps in earth science. In arXiv [cs.IR].

[4] Bozkurt, S., Alkim, E., Banerjee, I., & Rubin, D. L. (2019). Automated detection of measurements and their descriptors in radiology reports using a hybrid natural language processing algorithm. Journal of Digital Imaging, 32(4), 544–553.

[5] Department for environment, food and rural affairs Nobel house 17 smith square London SW1P 3JR telephone: 020 7238 6000 website: Www.Defra.Gov.Uk. (n.d.). Gov.Uk. Retrieved February 2, 2022

[6] Chen, X., Shuai, C., Wu, Y., & Zhang, Y. (2020). Analysis on the carbon emission peaks of China's industrial, building, transport, and agricultural sectors. The Science of the Total Environment, 709(135768), 135768.

[7] Dumortier, J., Hayes, D. J., Carriquiry, M., Dong, F., Du, X., Elobeid, A., Fabiosa, J. F., & Tokgoz, S. (2011). Sensitivity of carbon emission estimates from indirect land-use change. Applied Economic Perspectives and Policy, 33(3), 428–448.

[8] Gryparis, E., Papadopoulos, P., Leligou, H. C., & Psomopoulos, C. S. (2020). Electricity demand and carbon emission in power generation under high penetration of electric vehicles. A European Union perspective. Energy Reports, 6, 475–486.

[9] Samaras, C., Tsokolis, D., Toffolo, S., Magra, G., Ntziachristos, L., & Samaras, Z. (2018). Improving fuel consumption and CO2 emissions calculations in urban areas by coupling a dynamic micro traffic model with an instantaneous emissions model. Transportation Research. Part D, Transport and Environment, 65, 772–783.

[10] Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., & Gao, J. (2020). Deep learning based text classification: A comprehensive review. In arXiv [cs.CL].

[11] Malek, A., & Kumarasan, K. K. (2019). Design and development of a carbon footprint calculation model for Universiti Tenaga Nasional. International Journal of Recent Technology and Engineering, 8(4), 6236–6239.

[12] Shang, Mei & Geng, Haochen. (2021). A study on carbon emission calculation of residential buildings based on whole life cycle evaluation. E3S Web of Conferences. 261. 04013. 10.1051/e3sconf/202126104013.

[13] Zhang, S. (2021). Sentiment classification of news text data using intelligent model. Frontiers in Psychology, 12, 758967.

[14] Luo, X. (2021). Efficient English text classification using selected Machine Learning Techniques. Alexandria Engineering Journal, 60(3), 3401–3409.

11.GLOSSARY

- Natural language processing
- Recurrent Neural Network (RNN)
- Deep learning
- Emission Factor (EF)
- Consumption Unit (CU)
- Regular expression (Regax)

12.APPENDICES

Appendix A. Unit Conversion test

Appendix B. Application UI developed