Universiteit Leiden

Master Computer Science

Carbon Footprint Information Extraction from Businesses Annual Reports

Name:	Maria Ioanna Kafetzaki
Student ID:	s3001210
Date:	14/07/2023
Specialisation:	Artificial Intelligence
1st supervisor:	Prof. Verberne S.
2nd supervisor:	Prof. Spruit, M.R.

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

Climate change plays a crucial role in our lives. Therefore, each business is obligated to report its annual emissions in a sustainability report. The goal of this paper is to extract this information from the annual reports, in portrait or landscape orientation PDF format. This information can be in sentences or tables. For the sentences, we develop a hierarchical classification model that extracts the sentences labelled based on their context to "Carbon Footprint", "Reduction of Carbon Emissions" and "Target". These labels refer to the amounts or the reduction of the emissions or the targets that a company has respectively. Also, the models classified the sentences into the Scope the sentences contained based on the ESG protocol. We annotated 11 annual reports from companies from different industries to create and develop a method to tokenize the sentences to create an efficient dataset for a machine-learning model. We use and compare Multinomial Naive Bayes, Random Forest Classifier, Support Vector Classifier and BERT model in every step. We conclude that the BERT model and using the ESG-BERT model to keep the most relevant sentences for our problem is the best-performing classifier. In addition, enriching our dataset using the chatGPT model improves its performance. Regarding the Table extraction, we use and compare two different table detection models and then we compare again the same models for the classification. Multinomial Naive Bayes is our suggestion without using TF-IDF.

Keywords: Information Extraction, Carbon Emissions, Business Annual Reports, Table extraction, Multinomial Naive Bayes, Random Forest Classifier, Support Vector Classifier, BERT

Acknowledgements

I would like to foremost thank my supervisor from Leiden University, Professor Verberne Suzan, for her guidance, feedback and useful comments throughout this research. I would like also to thank her for always showing enthusiasm for this project and trying to find new ideas to inspire me. Also, I would like to thank my second supervisor Professor Spruit Marco for his valuable comments. Another thank you to the company where I work e-on Integration S.A and specifically to my manager Louropoulos Alexandros and my colleague Makri Marina for her guidance in the environmental data.
Finally, I would like to thank my family for their support and my boyfriend Barmpi Vasileio for his support throughout this research.

Contents

1 Introduction

1.1 Motivation and definition of the problem

Nowadays, the environment and climate change are crucial subjects for the whole planet. As carbon emissions are extremely increased year by year, the planet faces a deep crisis and therefore humanity deals with problems that are related to this, such as air pollution, increase in temperature and even energy crisis. Thus, companies should decrease their emissions in order to revert these circumstances, and they are obliged to create an annual sustainability report of their carbon emissions and their actions for emission reductions in accordance to the Paris Agreement.[1]
These data are not only useful for reporting but also for consulting, making smarter business decisions, starting new research projects, and accurately assessing their risk as well as for the governments to check the progress in each country, including their carbon footprint globally, because that will contribute to implementing sustainable solutions to prevent the rapid climate change while maintaining growth. In order to collect and evaluate the relevant information from the annual reports, there is a need for a data extraction pipeline.
As manual data extraction is complex and time-consuming work, the automation of this work is a relevant topic for companies. In this project, we will deal with the aforementioned automation of data extraction by sentence extraction and the classification of carbon emission as mentioned in three main categories:

- Carbon footprint
- Reduction of carbon emissions
- Targets for emission reduction

For each of the categories, the sentences found will be classified into three Scopes, based on the type of production of carbon emissions: direct, indirect or within the supply chain of the company. This split in Scopes is based on the ESG protocol [2].
There is also a need to deal with table extraction, as the majority of numerical data on carbon emissions is found in company tables.
Therefore, the main research question that we need to address is: **To what extent is it possible to extract environmental data from businesses' annual reports?**
Consequently, the main challenges of this project include data annotation of public annual sustainability reports of companies in various fields, the detection of all related sentences regarding a company's emissions and the classification of these into three basic categories and their Scopes using hierarchical classification. Furthermore, the specific amounts of carbon emissions will be dealt with using a different methodology that enables the extraction of data in tabular format.
Based on the above-provided explanations, the main research question can be addressed by addressing the following sub-research questions:

- What kind of information can we extract from an annual report of a company relating to carbon emissions?
- How can we produce quality data for this purpose?
- Which models are the most effective for this problem?

1.2 Contributions

The main contribution of this research is that a model for the classification of carbon emissions data into scopes is designed. Even though the classification has been done into several fields, this field has not been classified before. There are also more contributions such as:

- The usage of PDFs as data, that include unstructured text and its transformation into structured text.
- The annotation of almost six thousand sentences with carbon emission labels based on ESG categories creates a dataset for training machine learning models.
- The classification of free text, meaning that there are several ways for someone to write the company's carbon emissions, even though there are some rules of what they need to write. In many cases, there are not all the categories of carbon emissions, while in others there is no free text but just a table or an image.
- The usage of ChatGPT [1] as a tool to enrich your data and the comparison of this method as a method to balance the data.

2 Background and related work

This chapter includes the theoretical background of our research review. In this section, we will provide an overview of the methods that we will follow in this paper. Moreover, we will provide an overview of the models that we use for the machine learning tasks but also for balancing methods and we will describe the evaluation metrics. Also, a review of related work in every method we use, which includes studies that have used machine learning classifiers and information extraction techniques, is contained.

2.1 Text extraction and text classification

2.1.1 Text extraction from PDF

Text extraction from PDF files is a demanding task. The reason is that PDF is a layout-based format which specifies the fonts and positions of the individual characters rather than the semantic units of the text (e.g., words or paragraphs) and their role in the document (e.g., body text or caption) [3]. Although there is not much research for this task, as researchers face it as an engineering task, a lot of methods have been developed on this topic, as it is challenging to find the word order, reading order or paragraph boundaries, which are characteristics that are crucial for the correct extraction. Bast and Korzen [3] presented their model for text extraction, named Icecite, and compared it with fourteen other models, like PdfMiner[2] or pdf-extract. They found that Icecite outperforms all other models in finding word order, reading order and paragraph boundaries but still has some limitations because of the ruled-based approach. On the other hand, PdfMiner is a tool that extracts the text from PDF files into txt files by analyzing the text and trying to find the correct paragraphs and lines in it. Artifex Software has developed another tool named PyMuPDF[3], which is a tool that reads, edits or converts to other formats a PDF file, such as HTML, SVG, PDF, and CBZ.

[1] https://openai.com/blog/chatgpt
[2] https://github.com/euske/pdfminer#readme
[3] https://mupdf.com/

2.1.2 Text classification

Text classification is a natural language processing task that involves assigning predefined labels to textual documents based on their content. It is actually a supervised machine learning model, and its goal is to automatically analyze and organize large volumes of textual data, enabling efficient retrieval, information extraction, and decision-making processes. Text classification algorithms categorize documents [4], find spam emails [5] or develop a sentiment analysis model using Twitter comments [6].

2.1.3 Hierarchical text classification

In most cases, classification tasks are produced in a flattened structure, resulting in just a label or multiple labels for sentences. On the other hand, in hierarchical classification, the models are annotated based on their hierarchical dependency, meaning that the sentences are assigned to a class that is contained in a class sequence [7]. Koller and Sahami [8] proposed a hierarchical classification that trained Bayesian Classifiers for each step to classify documents and divided the problem into smaller problems in order to handle the issue of having hundreds of classes and thousands of features. They compared it with a flattened classifier and they found that a hierarchy classifier results in higher overall accuracy. Ruiz and Srinivasan [9] found that a text classification procedure capable of exploiting the conceptual connections between categories is more effective than a model that is not designed to exploit such information leading to the reduction of the computational complexity. In addition, hierarchical classification allows for a more organized and structured representation of the document corpus. It enables the creation of a taxonomy of categories, which can handle large-scale documents by grouping related categories together at different levels of the hierarchy and therefore create for each group of labels more manageable sub-tasks [10]. As carbon footprint information can be divided from large categories into smaller ones, this structure can give us advantages, not only with respect to the implementation time but also in the learning process, as in every step, the most relevant information of each sentence is useful for the learning process [11].

2.2 Models

In this project, the classification is starting by showing the related and unrelated sentences based on carbon emissions content. For this task, different models are used:

2.2.1 Random Forest Classifier (RF)

Random Forest Classifier is a machine learning algorithm that is used for both classification and regression tasks. It is an ensemble method that combines multiple decision trees to make predictions. Random Forest Classifier consists of individual decision trees that operate as an ensemble. Each decision tree is built using a random subset of the training data and a random subset of the input features. This randomness helps in creating diverse and uncorrelated trees, which collectively make more accurate predictions.
Moreover, the algorithm creates each decision tree by using a technique named bagging. With bagging the data are sampled with replacement and therefore multiple subsets of data are created to train each tree. Each decision tree independently predicts the class and the final output is the class selected by most trees.
Random Forest Classifier is not sensitive in overfitting because of the ensemble method, the sampling of the data but also because of the randomness in the feature selection. Instead of using all the features, each tree randomly selects a subset of features and trains the data. This randomness helps to reduce the correlation among the trees. Finally, Random Forest Classifier

can handle missing data as when a prediction is made, then it uses only the available features and the others are just ignored.[12] [13] [14]

2.2.2 Support Vector Classifier (SVC)

The Support Vector Machine is a non-probabilistic supervised learning algorithm that is used for regression or classification tasks in machine learning. Support Vector Machines are a really effective method when dealing with data with high dimensionality like text data.
The main idea behind Support Vector Classifiers is to find an optimal hyperplane that separates the data points in a high-dimensional feature space. The hyperplane depends on the number of features. If the futures are two, then the hyperplane is just a line while if the number of features is N, then we get a N - plane. This hyperplane tries to maximize the margin between the classes in order to generalise the results and robust the noise. The data points that are located closest to the hyperplane, called support vectors, play a crucial role in determining the optimal hyperplane. In other words, support vectors are used to find the optimal position and orientation of the hyperplane.

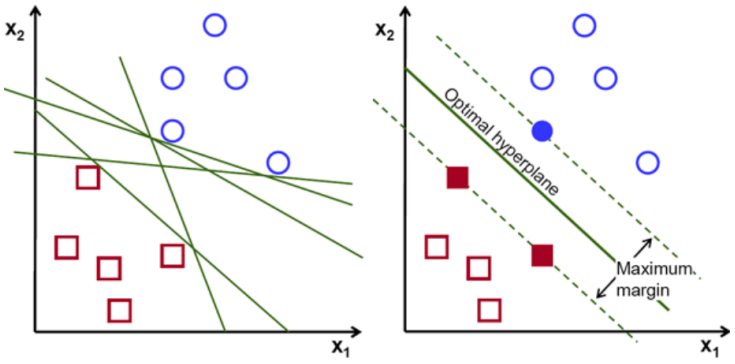


Figure 1: Possible hyperplanes and the optimal hyperplane found using support vectors. [4]

If there are non-linearly separable data, Support Vector Classifier implement other kernel functions like polynomial or radial basis function. these kernel functions transform the data in a higher dimensional space in that way to be easier to find a linearly separable hyperplane.
In order to get the best-separating hyperplane and generalize the model, we optimize the hinge loss function :

$$l = max(0, 1 - y^i(x^i - b))$$

where y^i and x^i are the ith instances in the training set and b the bias term. [15]
In the cost value, we also add a regularization parameter which balances the margin maximization and the loss. This is often described as the bias-variance trade-off [16]. Two regularization methods can be used: L1 and L2 regularization. Both methods are commonly used to prevent overfitting by diminishing the coefficients to zero or discarding them from the model. L1-regularization improves the generalization by shrinking the sum of the absolute values of the model coefficients, while L2-regularization uses a penalty function based on the sum of squares of the model coefficients. The main difference between L1 and L2 regularization is that in the former method, the regularized coefficients are more possible to be around zero and therefore this leads to sparser models. On the other hand in the latter method, the L2 penalty actually leads the coefficients to become smaller and finally to lead to non-sparse models. [16]

[4] https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

2.2.3 Multinomial Naive Bayes (MNB)

The Multinomial Naive Bayes algorithm is a probabilistic learning method that is widely used in Natural Language Processing (NLP), particularly when dealing with features that represent word frequencies or counts, like text classification.
The Multinomial Naive Bayes algorithm is based on the Bayes theorem which makes the assumption that the impact of a predictor's value (x) on a certain class (c) is unrelated to the values of the other predictions and calculates the probability of each of the predicted tags of a text for a sample as is shown in the following equation:

$$P(C|x) = \frac{P(x|C)P(C)}{P(x)}$$

where C is the set of class labels $C = \{c_1, c_2, ..., c_k\}$, x the feature vector that represents the document D. Because of the Naive Bayes assumption of feature independence, we allow factorizing the likelihood probability as:

$$P(x|c_i) = P(x_1|c_i) \cdot P(x_2|c_i) \cdot ... \cdot P(x_n|c_i) \text{ for i = 1,2,...,k}$$

where $x_1, x_2, ...x_n$ are the individual features in the vector x.
Next, we consider the Maximum Likelihood Estimation (MLE) for the probabilities P(x_i—c) based on the observed feature counts. The probability is estimated as:

$$P(x_i|c_i) = \frac{count(x_i, c_i) + \alpha}{\sum count(x|c_i)} + \alpha \cdot \text{vocabulary size for i = 1,2,...,k}$$

where count(x_i, c_i) represents the number of times feature x_i occurs in documents of class c_i, $\sum count(x|c_i)$ represents the total count of all features in class c_i, α is the smoothing parameter and vocabulary size represents the total number of unique features in the training data.
Now the probability P(C) is estimated as:

$$P(c_i) = \frac{count(c_i)}{\text{size of training set}}$$

where count(c_i) is the number of documents that belong to class c_i in the training data. The calculation of P(x), the probability of feature vector x, is not necessary for the classification decision since it is the same for all classes, so it can be ignored. Once the probabilities P(c_i—x) are calculated for each class c_i, the class with the highest probability is assigned as the predicted class for document D.
Some of the advantages of the Naive Bayes Classifier are that fewer data are necessary and it is better than other machine learning models when it is used for text classification tasks, such as spam detection, topic classification or sentiment analysis. It also works well with large feature spaces. [17]

2.2.4 BERT

Bidirectional Encoder Representations (BERT) is designed to pre-train deep bidirectional representations from the unlabeled text by joint conditioning on both the left and right context in all layers. BERT utilizes two steps: pre-training and fine-tuning. During the pre-training step, the model is trained on unlabeled data over different nlp tasks. In the second step, the BERT model is first initialized with all the parameters that are utilized in the best performance of the model in the pre-training. Then all of these parameters are fined-tuned using labelled data from the downstream tasks. Each task has separate fine-tuned models even if all of them had pre-trained with the same parameters.

The model's architecture is a multi-layer bidirectional transformer encoder. This architecture has been wildly successful in a variety of tasks in NLP. They compute vector-space representations of natural language that are suitable for use in deep-learning models. The BERT family of models uses the Transformer encoder architecture to process each token of input text in the full context of all tokens before and after, hence the name: Bidirectional Encoder Representations from Transformers. [18] This construction is shown in the following Figure:

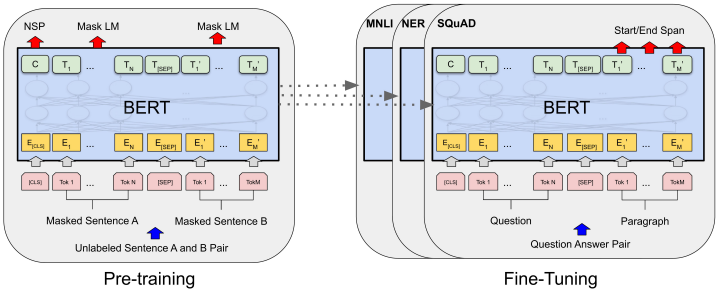


Figure 2: Bert Base architecture.[5]

BERT has achieved remarkable performance on a wide range of NLP benchmarks, surpassing previous state-of-the-art models. Its ability to capture contextual information and leverage large-scale pre-trained representations has made it a popular choice in various NLP applications. [19]

2.3 Table detection and text extraction from tables

For this study, it is also important to extract text from tables as in many cases tables provide us with relevant information about our research topic. For this task, there are two different parts. The first one includes table detection while the second one is about text extraction from the tables.

Table detection is a challenging task in machine learning because of the varying layouts and different encoding [20]. Thus, we do not develop a method for this task but we test and compare two pre-trained models in order to find the tables in the PDF files.

DEtection TRansformer (DETR) is an object detection model, that predicts all objects at once and is trained end-to-end with a set loss function which performs bipartite matching between predicted and ground-truth objects. DETR simplifies the detection pipeline by dropping multiple hand-designed components that encode prior knowledge, like spatial anchors or non-maximal suppression. Its architecture consists of a set prediction loss that forces unique matching between predicted and ground truth boxes and an architecture that predicts the relation of (in a single pass) a set of objects and models [21]. In other words, DETR consists of convolutional layers followed by an encoder-decoder Transformer. PDF images are used as input, while the output of the model consists of boundaries within which the object of interest is detected. The model structure is viewed in detail in Figure 3.

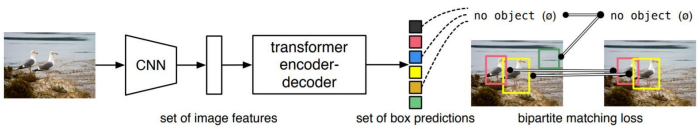


Figure 3: DERT architecture and its training method [21].

[5]https://paperswithcode.com/method/bert

In this paper, we are using the Microsoft Table Transformer and the TahaDouji Table Transformer:

1. The Microsoft Table transformer uses this architecture and is trained for two different problems: table detection and table structure recognition. The model is pre-trained with the PubTables-1M dataset [22] consisting f one million images of tables. In our research, we do not make use of table structure recognition functionalities, therefore only the table detection model is used. Apart from the input image sizes, width and height, a threshold that determines the value of model confidence the model can take in order to make a correct prediction, is imported. The value of this threshold for this problem will be 0.9 as there is an observation of high false positive predictions if the threshold is around 0.7, meaning that any input with an accuracy equal to or above 0.7 is classified as a table. The final output of the model is the value of the model confidence for each table but also the (x,y) minimum and maximum values, which will be used for the extraction of the table[6].

2. The TahaDouaji Table Transformer is based on detr-resnet-50 model [7], developed by Facebook. This model is trained on COCO 2017 object detection dataset[8] that includes 188 thousand annotated images. The TahaDouaji Table Transformer works exactly in the same way as the Microsoft Table Transformer, meaning that detects the table and the output is the place of the table and the certainty that in these coordinates, a table exists. It is worth noting that in this model, the user should also implement a threshold in certainty[9].

2.4 ChatGPT

ChatGPT is a Generative Pre-Trained Transformer language model developed by OpenAI. ChatGPT is trained using a huge dataset containing data from the internet, and therefore the model can learn patterns, grammar, vocabulary and context from different sources, creating divergent information. It is a language model with 175 billion parameters and it has strong performance on many NLP Tasks such as question-answer and reading comprehension. Tom Brown et al test its performance in a few-shot setting, meaning that the model can adapt and generalize from a small amount of labelled data, without any gradient updates or fine-tuning techniques. The authors conclude that ChatGPT has remarkable results in NLP tasks and in some cases, it can reach the performance of other state-of-the-art-fine -tuned systems [23]. Due to the above remarks, this research greatly benefits from this tool by using it for data augmentation.

2.5 Balancing methods

Here, we describe the methods used for balancing the data. Often real-world data are not equally classified and we need to address this problem, since in machine learning tasks when the data is unbalanced, this will lead models to predict only the majority class accurately, while the minority classes will lead to high errors.

2.5.1 Synthetic Minority Over-Sampling Technique (SMOTE)

SMOTE is an oversampling technique to deal with the class imbalance problem in machine learning tasks. It was proposed by Nitesh V. Chawla, et al. in 2002 [24].

[6]https://huggingface.co/microsoft/table-transformer-detection
[7]https://huggingface.co/facebook/detr-resnet-50
[8]https://cocodataset.org/download
[9]https://huggingface.co/TahaDouaji/detr-doc-table-detection

SMOTE approaches the problem by over-sampling the minority class by creating 'synthetic' examples rather than oversampling by replacement. In other words, the minority class is over-sampled by taking samples and introducing synthetic examples along the line segments joining any of the k minority class nearest neighbours. These samples are generated in the following way:

1. Take the difference between the sample under consideration of its k nearest neighbours, where k is the number of the neighbours that are considered.

2. Multiply this difference with a random number between 0 and 1.

This allows the selection of a random point between two specific features and the selection is more generalized. This procedure is repeated until the desired level of oversampling is achieved[10].[24]

2.5.2 Random Over-Sampling Technique

Random Over-Sampling is a simple and straightforward technique used to solve the problem of class imbalance in machine learning. In the class imbalanced dataset, the samples in the minority class are simply duplicated to increase the number of documents that exist in this class. The duplication process is repeated until the desired level of oversampling or class balance is achieved. Even though Random Sampling is a simple and easily utilised method, it may lead to overfitting if the majority class is much larger than the minority, and the model cannot predict new instances for the minority class as it had learnt only to predict the trained instances. [25]

2.5.3 Near Miss

Near Miss is an under-sampling technique used to solve the problem of class imbalance in machine learning. Instead of the other two methods described below, SMOTE 2.5.1 and Random Over-Sampling 2.5.2, Near Miss aims to reduce the number of samples in the majority class by selecting a subset of them. The selection is based on the samples that exist in the minority class as the models try to select the samples from the majority class that are more similar to those from the minority class. [25]
Thus, the algorithm works by measuring the distances of the similarities between the two classes and finding the closest from the majority to the minority class. All the other samples are ignored from the training set. There are three different ways to measure this distance, and therefore Near Miss algorithm has three different versions:

1. **Near miss version 1**: In this version, the majority class samples that are selected, have the shortest average Euclidean distance to the nearest minority class samples.

2. **Near miss version 2**: In this version, the majority class samples that are selected, have the farthest average Euclidean distance to the nearest minority class samples.

3. **Near miss version 3**: In this version, the algorithm considers the average distance from the k nearest neighbours from the minority class, where k is the number of selected neighbours, to select the instances from the majority class

2.6 Evaluation metrics

For the evaluation of our models, we use four different metrics: precision, recall, F1-score and accuracy[11].

[10] https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/
[11] https://en.wikipedia.org/wiki/Precision_and_recall

Precision

The precision measure describes how many instances assigned to a certain class were classified correctly. In other words, the precision of a class is the fraction of the number of true positives (tp), which with the definition "true positives" we describe the number of items that are correctly labelled to the class, divided by the number of instances that were predicted by the model to belong to the class, i.e the sum of true positives and false positives (fp), where "false positives" is the number of instances that were wrongly predicted to belong to this class. Thus, the precision of the model is the sum of the precision of the class divided by the number of classes and the equation that describes precision is:

$$precision = \sum_{i=1}^{k} \frac{tp}{tp + fp} \quad \text{where k is the number of classes}$$

Recall

The recall measures how many of the occurrences of a class are predicted correctly. In other words, recall is the fraction of the true positives divided by the number of occurrences that truly belong to the class, i.e. the sum of true positives and false negatives (fn), where "false negatives" is the number of instances of a class that predicted to belong in other class wrongly. Therefore, the recall of the model is the sum of the recall of the class divided by the number of classes as described by the equation:

$$recall = \sum_{i=1}^{k} \frac{tp}{tp + fn} \quad \text{where k is the number of classes}$$

F1-score

F1-score is the harmonic mean of recall and precision. F1-score can be 0 to 1 and it is actually the combination of precision and recall. In addition, the F1-score of the model is the sum of the F1-score of the class divided by the number of classes and it is described as:

$$F1 - score = \sum_{i=1}^{k} 2\frac{precision \cdot recall}{precision + recall} = \frac{2tp}{2tp + fp + fn} \quad \text{where k is the number of classes}$$

Accuracy

Accuracy describes the overall performance of the model. It is the number of truly predicted labels to a class divided by the instances of a class. In other words, the accuracy of the class is the number of true positives and true negatives (tn), where "true negatives" is the number of correctly not predicted in a class, divided by the number of instances in the class. The accuracy of the model is the sum of the accuracy of the class divided by the number of classes and is expressed as an equation as follows:

$$accuracy = \sum_{i=1}^{k} \frac{tp + tn}{tp + tn + fp + fn} \quad \text{where k is the number of classes}$$

2.7 Related work

Over the past few decades, a lot of studies have been made in automatic information extraction from unstructured or semi-structured documents in a variety of fields like web pages [26, 27] or paper documents [28]. The most relevant research to this project is the information extraction of financial data [29, 30], which focuses on the extraction of main contents or segment names from business annual reports. In these papers, the authors attempt to give a solution for the avoidance of manual extraction but also the easier usage of the data for a better understanding of a company's progress. Moreover, there are studies for information extraction that focus on long documents with complex contents like graphs, tables or unstructured formats like PDF files [31] in which the authors create new datasets and try to create a pipeline that uses pre-trained models like BERT for information extraction of real-world business data.
In addition, there are studies that focus on information extraction from research articles by using a combination of text-mining techniques in order to find patterns and trends in huge datasets in order to extract interesting information and find similarities in them. [32]. There is interesting research for analysis of supervised text classification algorithms on corporate sustainability reports that tests and compares some machine learning models, such as the Naive Bayes Classifier and Decision Tree for classifying sustainability reports into sections. In this research, the authors answer the following questions: 1) Which techniques should be used to be effective in the classification under the appropriate categories and 2) How good are the results compared with the manual method? The authors recommend using dimensionality reduction for preprocessing and they suggest using Neural Networks and Probabilistic methods instead of Decision tree models [33].
Another important feature that we need to explore is tabular information extraction. A lot of studies have been made in recent years related to tabular format and the way that data can be extracted. As the documents are PDF files the first step is table detection, which the authors in [34] achieve by using not only the graphic ruling lines and the white spaces but also the page columns in order to find the tables in a PDF. Once a table has been detected, the second part is extracting the data from it. An interesting idea is the classification and information extraction for tabular data in images, as proposed by Amir Riad et al. [35]

3 Data

This research focuses on the real-world data from companies and more specifically, on the annual report that every company should publish, regarding their progress into sustainability, such as environment, waste, digital responsibility, employees etc. These reports are published and are accessible to anyone to read. In this paper, we use reports from eleven different companies, in different industrial fields, such as retail, telecommunication, maritime, gas production, entertainment, banking, food industry, construction, vehicle manufacturing and airlines in order to enrich the vocabulary used in the learning process. These reports can have portrait orientation meaning that they have vertical pages or in landscape orientation, which means that the PDF has horizontal pages. Also, each PDF contains text, graphs and images in different formats, which makes text extraction from PDFs challenging. Table 1 provides an overview of the documents in our collection.

Company Name	Year	Format	No Of Pages	No of Words
Apple[36]	2022	Portrait orientation	128	51286
Microsoft[37]	2021	Landscape orientation	119	50400
Coca-Cola[38]	2021	Landscape orientation	86	41422
NIKE[39]	2021	Portrait orientation	184	53118
IKEA[40]	2021	Landscape orientation	32	17900
Samsung[41]	2021	Landscape orientation	111	63535
KLM[42]	2021	Portrait orientation	206	85644
Van Oord[43]	2021	Portrait orientation	138	50401
Amazon[44]	2021	Landscape orientation	101	47652
Meta[45]	2021	Landscape orientation	114	28117
Mercedes-Benz[46]	2021	Portrait orientation	289	133265

Table 1: Introduce the PDF files, giving the name of the company, the year that the report was published, the number of pages, words and the format of each PDF file.

3.1 Data annotation

This thesis aims to extract environmental data from the annual reports of businesses. To achieve that, manual data annotation is crucial. Data Annotation is the procedure of labelling the data to use it as input for a machine learning model. The annotation should be done by sentences, provided that this is a label-by-sequence classification problem.

3.1.1 Preparation of PDF file for the tokenization

Before the data are annotated, the preparation of them is needed. To begin with, some pre-processing in the PDF files should be done, in order to create more structured data. The first part includes the removal of images and graphs from the reports so that the document becomes simplified. Table detection is the next step in our process. We initially experimented with NLTK[12], spaCy[13], or simple sentence tokenization using patterns, but none of these gave the desired results. All these three methods cannot detect the tabular format of the data and it takes them as free text and consequently, they return the sentences wrongly. Due to the fact that tabular data cannot be used as free text, we need to create two different tasks. One will be the free text extraction while the second one refers to information extraction from tables. Thus, we proposed our own pipeline which is described below:

1. Transform every page into an image
2. Detect all the tabular data that exist in the document using the table detection model, described in section 2.3, which returns the boundaries of the table (in pixels) on the page. In this step, both models that we discussed in the background section are tested. We observed that Microsoft Table Extraction had higher model confidence in table detection. By trial and error, we identified that when the threshold was set at 0.9, the model was unable to detect more than one table per page. When lowering the threshold, not only

[12]https://www.nltk.org/
[13]https://spacy.io/api/doc

did it fail to detect multiple tables but also misclassified free text as tables. On the other hand, the TahaDouaji model achieved the desired outcome of detecting tables by using a threshold of 0.7 while also having some, but fewer, misclassifications of free text as tables. As far as results go, in our datasets, out of 317 tables, all 317 were accurately detected and 30 were free text falsely classified as tables. This results in an accuracy of around 90%

3. Remove the table from the page, by cropping the boundaries notated by the TahaDouaji model's outputs.

4. Extract the text that the table contains.

5. Remove the sentences that are contained in the table, separating them from the rest of the free text in the PDF. By doing so, we have created two datasets: one containing the table data used for table information extraction, and one with free text used for hierarchical classification.

After that, the text that has remained is cleaner than before, and therefore the tokenization is easier because the information that is contained in the data is only from the free text. It is worth noting that the companies' names are replaced with the word 'company' in order to generalise the sentences. The final part is the tokenization for which we use the NLTK library. Although the quality of the sentences' split is high, there are some text sequences that the tokenizer should have split into multiple sentences and fails to do so. The below scenarios showcase how this can be problematic for the goal of our project and the implemented solutions:

1. If there is a change of line and the first letter is uppercase but there is no full stop at the end of the sentence. In most cases, this happens in the contents or in some images that are not deleted from the model. A good example is:

   "Climate Change
   Resources
   Smarter Chemistry
   Engagement"

   which is returned as a sentence.

   In order to solve this problem the first letter in the next line is checked. If it is uppercase, a full stop is placed at the end of the sentence. This solution has a shortcoming if the first word in the next line is a word that is written with an uppercase letter as the first letter, like a name, but as a general rule of thumb, the tokenizer produces better quality sentences.

2. If there is a full stop that is not followed by a space and the first item after the dot is a letter and not a digit. For example, the following sentences need to split into two but only one is returned: "We find ways to consume energy more efficiently, and we seek out opportunities to transition to renewable sources that support our goal of 100 per cent renewable electricity across our operations and supply chain.With the renewable energy we source, we aim to achieve positive impacts."

   In order to solve this problem, the algorithm detects all the full stops in the text. If the next item after the full stop is a letter and not a digit, the algorithm places a space between them.

3. If there is more than one space and the next letter after that is upper but there is not a full stop between them. For example the sentence: "• Equity investment (3 per cent

of company-created projects) We invest capital in new solar PV or wind projects in some markets, matching the renewable energy generated with our energy use."

We solve this problem by counting the spaces between the words. If more spaces than one are detected, a full stop after the last letter before the first space is placed, and all the other spaces except one are deleted.

After this function is applied, the NLTK library is used for splitting the text into sentences. The number of sentences for each document is described in Table 2.

Company Name	**No Of Sentences**
Apple	3386
Microsoft	4162
Coca-Cola	4258
NIKE	5092
IKEA	1134
Samsung	6300
KLM	6184
Van Oord	3704
Amazon	2743
Meta	3161
Mercedes-Benz	8406

Table 2: Number of sentences for each document after applying the algorithm for solving the tokenization limitations and the NLTK library to split the sentences.

From all the sentences, we drop the ones that have 2 or fewer words, as we observe that in most cases these are the contents or titles. The sentiment behind this notion is that these word sequences, statistically, have a very rare chance of containing relevant data in the context of carbon emissions. Furthermore, we exclude sentences in which we count more digits than letters as there is a chance for a table to not be deleted and thus we try to exclude the sentences that are from a table to have more clean data.

3.1.2 Labelling explanation

The labels of this study were based on the definitions from the Greenhouse Gas Protocol [47]. Business reports provide a large amount of environmental information. In this project, we will specifically focus on the carbon footprint of a business, in the current year and previous years, as well as the targets that are defined to reduce it.

Before we categorized the labels into the categories of the carbon footprint, we needed to find the relevance of the subject sentences. As mentioned before, a company's report comprises a lot of sections, but we only need the carbon emissions section. Thus, the first labelling step is relevant and irrelevant sentences.
Next, the categorization in the main labels is needed. The main labels, as mentioned in introduction 1, are **the carbon footprint**, **the reduction of carbon footprint** and **the targets**.

Carbon Footprint is the amount of GHG emissions (carbon dioxide (CO_2), methane (CH_4) and nitrous oxide (N_2O)) calculated in CO_2 equivalent (CO_{2_e}) released into the atmosphere as a result of the activities of a particular organization. The second main label is the reduction of emissions, including all the actions that a business takes to reduce its emissions. Target refers to the goals that a company has set for the future regarding its emissions. The main goal, for most of the companies, is to have a net-zero carbon emission rate until 2050 based on the Paris Agreement[1]. Therefore, the annotation is based on the targets that a company has for each Scope but also on the company's main goal.
After the first labelling with 1 of 3 classes, all the sentences are also categorized into Scope 1, Scope 2 and Scope 3, but also a fourth category which includes a general sentence but relevant to the subject.

- Scope 1 emissions are direct emissions of a company, meaning the emissions that are released into the atmosphere from their activities. This category includes mobile combustion, fugitive emissions and stationary combustion. Mobile combustion refers to the fuel that a vehicle needs without electricity, which is reported in Scope 2. Fugitive emissions are the leaks of GHG emissions process, which are mainly the emissions produced by all the processes of a company. Stationary combustion regards the emissions that are released into the atmosphere for heating, electricity or other fuel production.

- Scope 2 emissions are indirect emissions from the generation of purchased energy, by a utility provider. In other words, all GHG emissions released into the atmosphere originate from the consumption of purchased electricity, steam, heat, and cooling.

- Scope 3 emissions are indirect emissions that are not included in Scope 2 and are referred to as the emissions that are included in the value chain of a company. This category is split into fifteen sub-categories but in this project, we are only concerned about the following, based on the companies that are chosen for the research:

  - Upstream activities - business travels
  - Employee commuting - travel from and to work
  - Franchises - companies that operate franchises and pay a fee to the franchiser
  - Used of sold products - concerning "in-use" products that are sold to the consumers. It measures the emissions resulting from product usage, even if it varies considerably

Table 3.1.2 shows the exact contents for each category based on the GRI standards. [48]

Scope	Contents
Scope 1	Generation of electricity, heating, cooling and steam Physical or chemical processing Transportation of materials, products, waste, workers and passengers Fugitive emissions
Scope 2	Generation of purchased or acquired electricity, heating, cooling and steam consumed by an organization
Scope 3	Other consequences of an organization's activities that occur from sources not owned or controlled by the organization Upstream and downstream emissions

Table 3: Content that should be marked by each Scope

3.1.3 Data annotation procedure

Starting from an annual report, every sentence that is relevant has to be found and annotated properly. The labelling procedure is done manually, reading every sentence and creating, for every step of the hierarchical classification, a label. Thus, these sentences should be marked by three labels. The first one is referred to the relevance of the sentence to the subject, so we have the relevant and irrelevant task. This is done because our data are imbalanced. If we do not split the sentences into relevant and irrelevant, we will have 5 classes with really imbalanced samples. Next, we annotate the sentence with the main category (Carbon footprint, Reduction of emissions, Target) while the third one is Scope 1, 2 and 3. Note that there are companies that report the reduction and the targets in general, so a sentence in these categories may take only one label and be marked as general. In Table 4, some examples for each label are presented.

Main Label **(Category)**	**Secondary Label** **(Subcategory)**	**Example**
Carbon Footprint	Scope 1	Of this total, about 120,000 mtCO2 were Scope 1 emissions at company's data centres.
Carbon Footprint	Scope 2	The system has an annual output of approximately 750,000 kWh/year of electricity, equivalent to about 15 per cent of all company's energy usage.
Carbon Footprint	Scope 3	The high consuming of our emissions are in Scope 3, more than 97%, which includes emissions from our supply chain, the lifecycle of our hardware products and devices travel, and other indirect sources.
Reduction of emissions	Scope 1	Scope 1 (direct) emissions were 33.2MtCO2e in 2021, a decrease of 20% from 41.7MtCO2e in 2020.
Reduction of emissions	Scope 2	We are working on implementing energy efficiency measures to reduce our Scope 2 emissions, such as installing LED lighting and improving insulation in our buildings.
Reduction of emissions	Scope 3	Since 2016, the climate footprint of our products used by customers at home has decreased by 4.9%.
Reduction of emissions	General	We reduced our Scope 1 and 2 emissions by 58,654 metric tons of carbon dioxide equivalents (mtCO2) in 2021.
Targets	Scope 1	By 2030, we aim to use initiatives to remove enough carbon dioxide from the atmosphere to cover the direct emissions.
Targets	Scope 2	We will increase the use of renewable electricity to run our offices to 100% by the end of 2023 and reduce GHG emissions per employee by 30% in 2025 compared with 2019.
Targets	Scope 3	The company has committed to reducing scope 3 emissions from our supply chain by 30% over the next 10 years.
Targets	General	We will reduce our Scope 1 and 2 emissions by the middle of the decade through energy efficiency work and reaching 100 per cent renewable energy by 2030

Table 4: Examples of sentences that will be annotated for every label.

It is worth noting that for all other steps, the sentences that have already been annotated as irrelevant, continue to have the same label in all other steps, meaning that an irrelevant sentence has labels: Irrelevant → Irrelevant → Irrelevant, because the model that we will design and it is described in section 4.2 might not be 100% efficient.

Regarding the Table extraction, we extract the tables using the pseudocode explained in section 3.1.1. After that, we have overall 317 tables. For each table, we extract the text and we annotate every extracted text as "Relevant" or "Irrelevant" based on the relevance of carbon emissions. For this annotation, we do not split the text into sentences, but we get the whole text as one observation.

4 Methods

In this section, we explain the procedure to create our models for the two different tasks.

4.1 Preprocessing

In this section, we describe the preprocessing which is done in order to be an efficient training set for our models. It is worth noting that all these processes are done for Multinomial Naive Bays, Support Vector Classifier and Random Forest Classifier models. For BERT this part is skipped as the BERT model needs a context and natural language text.

4.1.1 Preprocessing of the sentences

After the sentence creation, described in the previous section, these sentences need to be preprocessed. The first part is the removal of all the symbols and the punctuation as they do not have any special meaning for the problem and they appear frequently in the sentences. The numbers should be kept as they are essential for our research because they provide not only useful information for the subject but also we have three different scopes that are written by definition as 'scope 1','scope 2','scope 3', and our goal is to classify in this way. Secondly, all the capital letters are changed to lowercase. In the next step, we eliminate the URLs that may be in our data, and finally, the deletion of stopwords included in the NLTK library.

4.1.2 Prepare the data for the models

After the cleaning process, we need to tokenize the type of the words so the input of the model will be an integer and not a string. So, the first step is to replace each word with a unique integer. In this step, the CountVectorizer [14] is used that transforms a given text into a vector on the basis of the frequency of each word that occurs in the entire text.
Next, we use Term Frequency - Inverse Document Frequency (TF-IDF). This is a statistical method that aims to better define how important a word is for a document, taking into account the relation to other documents from the same corpus. This is performed by looking at how many times a word appears in a document while also paying attention to how many times the same word appears in other documents in the corpus. Its calculation is done as described in equations 1,2.

$$TF = \frac{\text{Number of repetitions of a word in a document}}{\text{Number of words in a document}} \tag{1}$$

[14] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html

$$IDF = \frac{\text{Number of documents}}{\text{Number of documents containing the word}} \tag{2}$$

Multiplying TF and IDF scores, we get the TF-IDF score. TF-IDF is a score which is applied to every word in every sentence in our dataset. And for every word, the TF-IDF value increases with every appearance of the word in a sentence but is gradually decreased with every appearance in other sentences[15].

4.2 Methodology procedure for the models

In this subsection, we outline the methodology used in the hierarchical classification. This hierarchy consists of three steps, the method and the classes that we followed to annotate the data. Therefore, the first step involves clarifying the relevant sentences, the second step is the categorization into the three main labels as has been referenced 3.1.2, and in the final step, we clarify the scope in which the sentence is included.

4.2.1 Classify the data into relevant and irrelevant

This is the first step in our experiment. In this step, we try to classify our data, meaning the sentences of each document into relevant and irrelevant sentences based on our annotation in section 3.1.3. Thus, we need to develop a binary classification model. Moreover, as the data are imbalanced, we use different techniques to balance the data. For this step, we use all the methods described in section 2.5. Specifically for the Near Miss method, we use version 2.
The step of relevant and irrelevant categorization is fundamental, because if we cannot find the relevant sentences then we cannot continue with our main goal. It is worth noting, that we also compare the binary classifier with the ESG-BERT classifier which is a BERT model, fine-tuned to return the annual report into 26 different categories. These categories are found in the table 12 and we use only the sentences that are characterized as GHG_Emissions by the model.

4.2.2 Classify the data into Carbon Footprint, Reduction Of Carbon Emissions, Target

Following the categorization of the sentences into relevant and irrelevant, we proceed to the second step, which actually performs a four-class classification. This happens because even if we want the three main labels to be classified, we need to also have an extra class for the irrelevant sentences. Again we will train the same models as the first step and compare the results. Also, we include our own idea for balancing the data, using the ChatGPT model and generating new sentences relevant to our problem by asking the following questions :

- Can you give me 100 sentences for carbon emissions related to amounts, reduction or targets in business?
- Can you rephrase these sentences into different unique sentences?

A small sample of these sentences is found in the appendix section 13.

[15]https://medium.com/analytics-vidhya/tf-idf-term-frequency-technique-easiest-explanation-for-text-classification-in-nlp-with-code-8ca3912e58c3

4.2.3 Classify the data into Scope 1, Scope 2, Scope 3

For this task, we use two different techniques. Firstly, we try to classify the data by continuing the hierarchical classification from the previous step, while in the second one, we try to develop a multi-label classifier, returning the sentences labelled with one or more Scopes. In other words, the former method follows the hierarchy of the previous steps. Therefore there is a need to include not only the three Scopes but also the irrelevant class and one more, the general one, as there are sentences that include information for two or more Scopes, which is already explained in the data annotation section 3.1.3. The hierarchy is described in Figure 4.

Hierarchy Classification Test 1

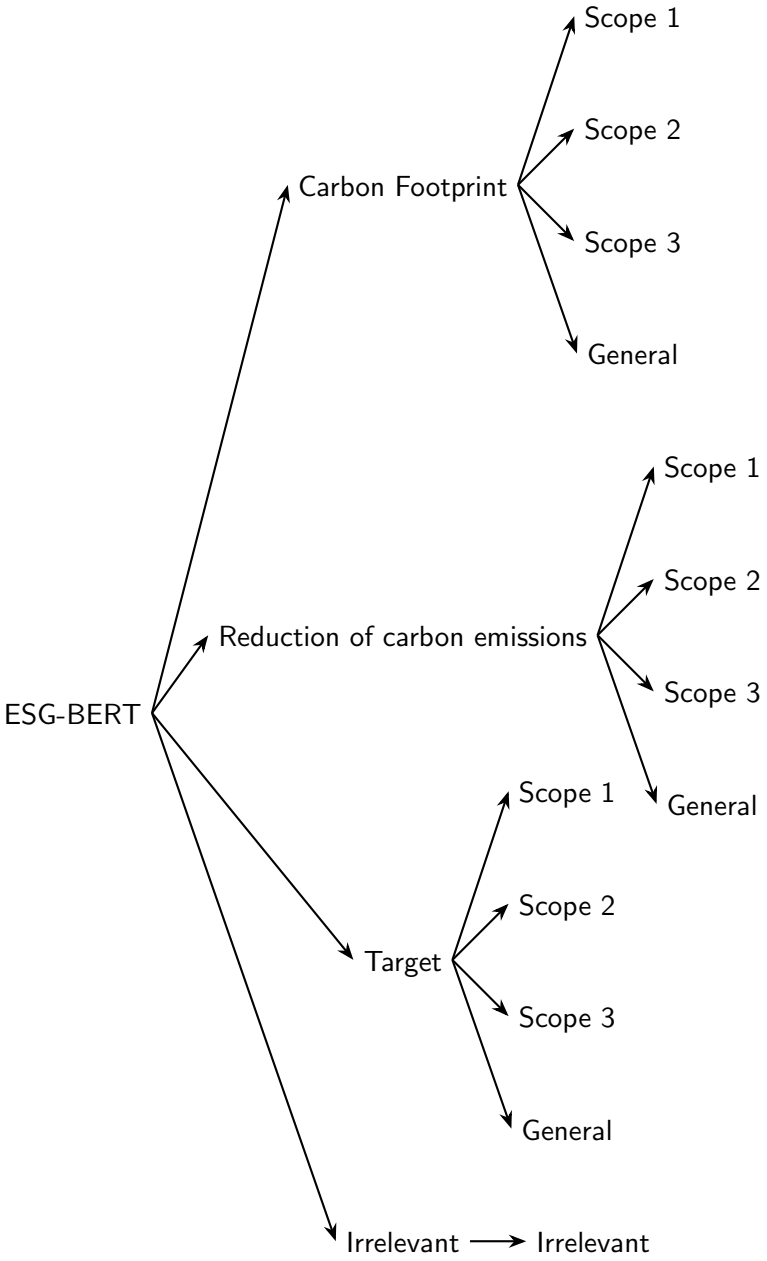


Figure 4: Hierarchy classification for the two labels together. The Figure describes the occasion in which there are no mistakes. However, as the accuracy is not 100% (there are false negatives) all the classes (Carbon footprint, Reduction of carbon emissions and Target) are connected with Irrelevant and also the Irrelevant (there are false positives) with the second classes (Scope 1, Scope 2, Scope 3 and General).

In the second trial, we take the two labels as independent, meaning that we do not continue the classification from the three previous labels. Instead of that, we classify the labels only as relevant and irrelevant and then we train a multi-label classifier that returns one or more Scopes for each sentence. Now the models become as described in Figures 4.2.3 and 6

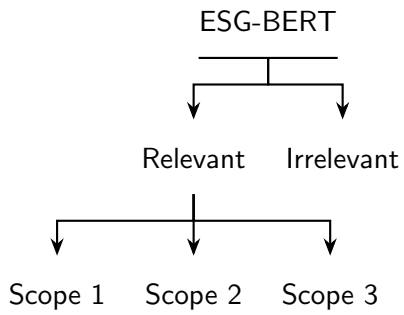
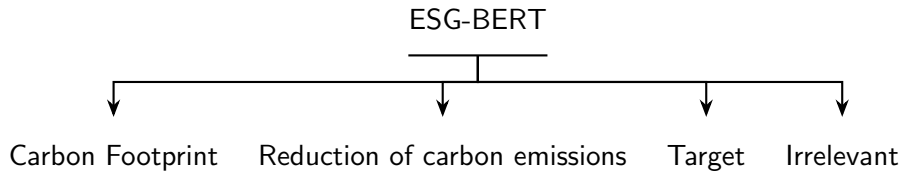


Figure 6: Multi label Classification for the second label. In the second step, we take only the relevant sentences. Of course, as the model is not 100 % accurate to predict the relevant and irrelevant, we take some truly irrelevant and classify them into the labels, Scope 1, Scope 2, and/or Scope 3. The Figure shows the desirable classification.

Hierarchy Classification Test 2



Classification for the main label.
Classification for the main label.

Figure 5: Classification for the main label.

4.2.4 Extract relevant tables

Regarding table extraction, we want to extract the relevant tables into table format. For this task we will use the same models as the categorization of the free text, but without using balancing methods, since the tables are only 317 overall. So, as the amount of data is really low, we do not use cross-validation per dataset, but cross-validation with 10 files split. Therefore, we use the following procedure:

- Extract text from the images that we have created in the preparation of the data procedure using pyMuPDF.
- Give labels to the tables as Relevant/Irrelevant
- Create a model that extracts the relevant tables.
- Extract the table as an Excel file using pandas.

It is worth noting, that we will test the models with or without the TF-IDF function because it may not be efficient for our models.

5 Experiments - Results

In this section, we present the different experiments for each step as it is explained in the methods section 4.2.

In this project, eleven different PDF files have been annotated and they have been used as training and test sets. In all experiments the training process has been done by using cross-validation split by document, meaning that in every iteration of the training, ten documents are the training set while the remaining is the test set.

5.1 Classify the data into relevant and irrelevant

Starting the classification, we observe that our dataset is unbalanced as is shown in the following graph:

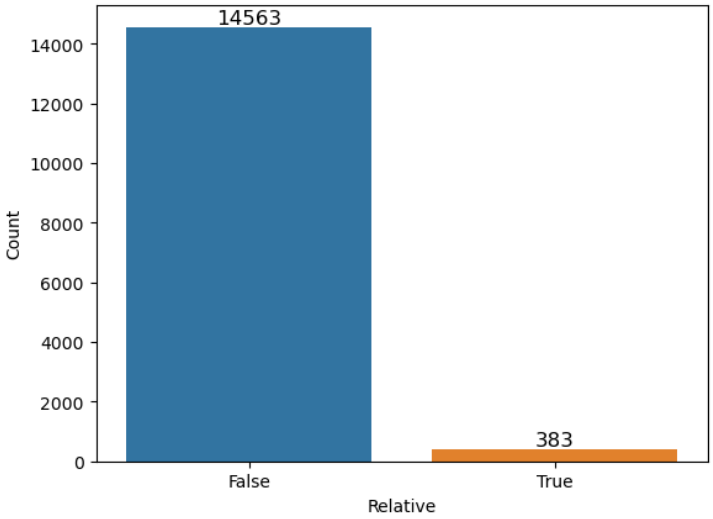


Figure 7: No of observations for the two classes in the training dataset. False represents the irrelevant while True is the relevant label.

Thus, different methods of balancing an unbalanced dataset will be used, meaning oversampling using the SMOTE and RandomOverSampling methods and undersampling using the NearMiss version 2 as they are offered from the sklearn library.
The training data become as :

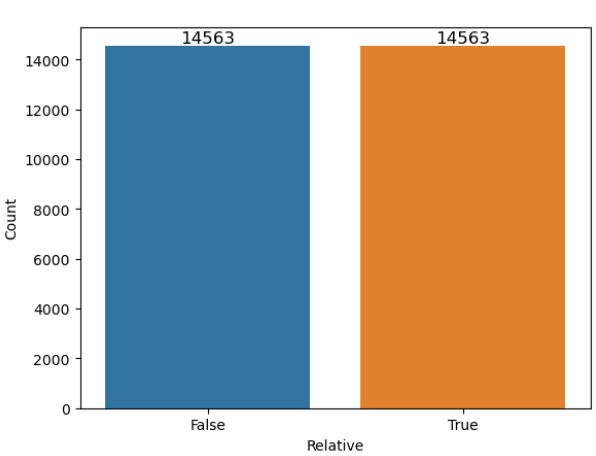


Figure 8: Over Sampling method - SMOTE

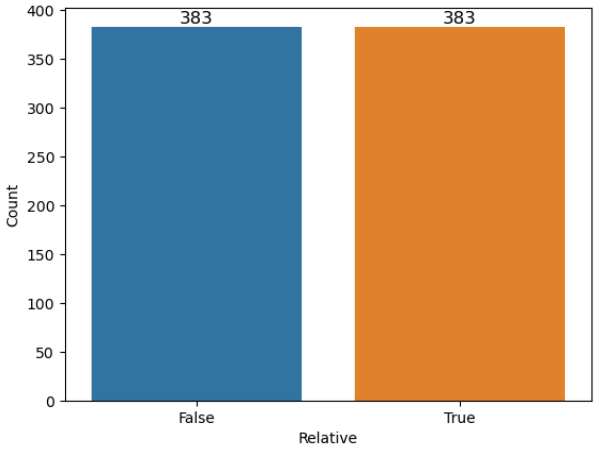


Figure 9: Under Sampling method - NearMiss v2

The evaluation metrics that will be used are precision and recall scores. Overall, the goal is

to maximize both precision and recall but emphasise recall maximisation, as if we do not find relative sentences we cannot continue to the next step, which is the classification of the main three labels introduced in section 1.
Starting the experiments with different models and different imbalanced methods, we get the following results:

Model	**Balance Method**	**Precision**	**Recall**
RF	None	0.00	0.00
RF	SMOTE	**0.84**	0.19
RF	Random Over Sampling	0.64	0.20
RF	Near Miss v2	0.03	0.94
MNB	None	0.00	0.00
MNB	SMOTE	0.05	0.35
MNB	Random Over Sampling	0.05	0.35
MNB	Near Miss v2	0.02	0.86
SVC	None	0.00	0.00
SVC	SMOTE	0.14	0.20
SVC	Random Over Sampling	0.64	0.20
SVC	Near Miss v2	0.02	**0.96**
BERT	None	0.00	0.00

Table 5: Experiments of different models with different methods for balancing the dataset and their precision and recall scores results for binary classification to relevant and irrelevant.

Even though the recall of the models is quite high, the precision does not respond to the desired goal. Observing the most frequent words in our dataset before using an imbalanced method, we can see that none of the keywords like 'scope', 'emissions' or 'reduction' are included in the sentences, as it is shown in Figure 10.
Moreover, if we take as an example the observation of the confusion matrices for a random document, for the models with the best recall as it is explained in Figure 20 and precision, which its confusion matrix is presented in Figure 19, respectively, as well as for one of the models without a balancing method, that we observe the classification in Figure 18, 8, we notice that even though the model with the near miss method returns most of the relevant sentences correctly, for the irrelevant class it returns a lot of the sentences as relevant, which it concludes to have more irrelevant than relevant sentences for the next step in the classification.
Therefore, the ESG-BERT model will be used. We take only the sentences that are annotated from the model as GHG_Emissions. As the model does not have 100% accuracy, there are some sentences that exist in other categories such as Air_Quality or Energy_Management but there are also irrelevant sentences of the subject in the GHG_Emissions category.
Taking the same example that we presented in the previous tests, we observe that if we take only the sentences that are mentioned as GHG_Emissions we have a set of 82 sentences. Out of

Figure 10: Most frequent words in the dataset before using an imbalanced method.

which, 52 are relevant based on our annotation, which is the exact number that is relevant for the model as well, while 30 are classified in this category, which is irrelevant. So we take only 30 classified wrong sentences, as in the other models we have had worse results, as can be seen in the Figure 5.1.

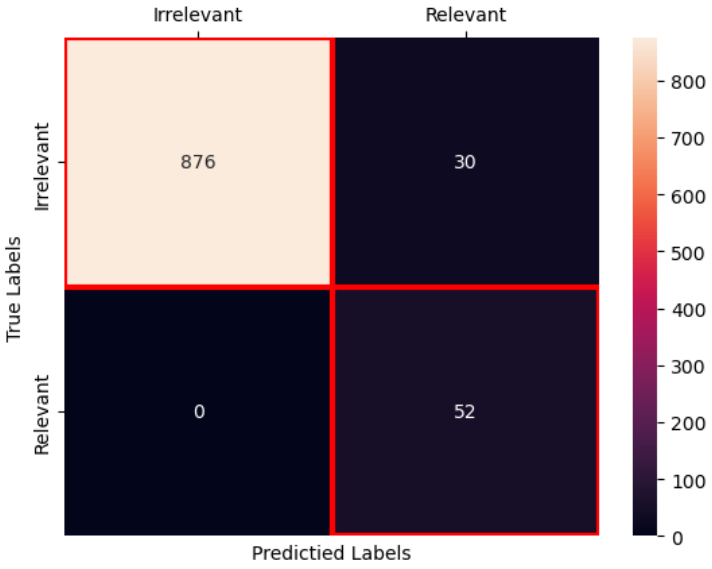


Figure 11: Confusion Matrix for ESG-BERT model classified in ESG_Emissions labels, which is the relevant class and the irrelevant, which includes the sentences classified in another of 25 categories, for a random of our documents.

Moreover, using the ESG-BERT, the training data includes 11 documents and the most frequent words in the training dataset have changed, as is depicted in the Figure 12.

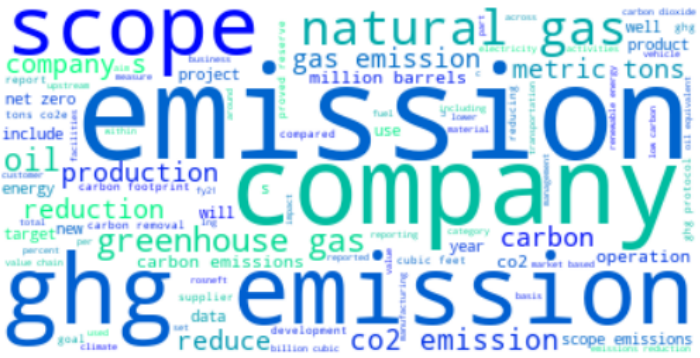


Figure 12: Most frequent words in the dataset after the implementation of the ESG BERT model and the selection of the GHG emissions category.

5.2 Classify the data into Carbon Footprint, Reduction Of Carbon Emissions, Target

Continuing from the previous stage, the classification does not only have the three labels of "Target", "Reduction Of Carbon Emissions" and "Carbon Footprint" but there is a fourth category created, labelled as "Irrelevant". Based on that, the problem is converted into a classification task with four classes instead of two. Again, the same models will be tested. It is worth noting that in this step the tests are done not only using a method for imbalanced data but also the enrichment in our dataset which will be done using ChatGPT.
Now we utilize the same models as previously, including and excluding the sentences that are generated from ChatGPT in the training set. In our experiments, cross-validation will be used in order to prevent overfitting. In every iteration, 10 datasets will be the training set with or without the ChatGPT sentences while the remaining will be the test set. For the oversampling or undersampling methods, SMOTE and NearMiss version 2 will be used respectively, without the new sentences from ChatGPT. For the evaluation of the models, precision, recall and F1 scores will be calculated.
If we do not use the ESG-BERT and we try to classify the sentences using only our annotation for relevant and irrelevant sentences and now the relevant sentences are categorised into the three labels of "Target", "Reduction Of Carbon Emissions" and "Carbon Footprint", we observe in the appendix section the confusion matrix of a random document that using the BERT model, all the sentences are again predicted as irrelevant 22. On the other hand, training the BERT using first the ESG-BERT classifier and taking as data the sentences labelled as GHG_Emissions we notice that the results are remarkably improved in Figure 23. Therefore, we test all the models using the ESG-BERT first and take the sentences that are annotated as GHG_Emissions as the training set.
The results are shown in the Table 6.

5.3 Classify the data into Scope 1, Scope 2, Scope 3

For the final step, we try two different methods as we talked about in the methodology section 4.2.

5.3.1 Continuing the hierarchy classification

In the first method, we use a flattened classification for the final step. In other words, we use 5 different labels, Scope 1, Scope 2, Scope 3, General and Irrelevant, as described in section 4.2.3,

Model	Balance Method	Precision	Recall	F1-score
RF	None	0.70	0.64	0.62
RF	SMOTE	0.68	0.65	0.63
RF	Near Miss v2	0.68	0.65	0.63
RF	ChatGPT	0.71	0.69	0.67
MNB	None	0.61	0.54	0.52
MNB	SMOTE	0.68	0.53	0.53
MNB	Near Miss v2	0.65	0.52	0.52
MNB	ChatGPT	0.67	0.61	0.61
SVC	None	0.67	0.61	0.59
SVC	SMOTE	0.65	0.60	0.58
SVC	Near Miss v2	0.66	0.63	0.61
SVC	ChatGPT	0.69	0.66	0.64
BERT	None	0.86	0.85	0.85
BERT	SMOTE	0.86	0.85	0.85
BERT	Near Miss v1	0.66	0.63	0.61
BERT	**ChatGPT**	**0.92**	**0.91**	**0.91**

Table 6: Experiments of different models for multi-class classification for the second step with different methods for balancing the dataset and their precision, recall and F1-score results.

and we try to classify the sentences from the labels Carbon Footprint, Reduction of carbon emissions and Target into the final categories. Thus the problem can be translated as a 5-class classification problem. The models that we used in the previous steps will be tested in this step as well. It is worth noting that even though the desirable model is to extract only scopes, there are also irrelevant labels in this step as there are also irrelevant sentences that perhaps exist in the sentences that the model could not recognize as irrelevant in the previous step. So we get a hierarchical classification, in which the final step is the categorization of the scopes and the general sentences, as described, in the data annotation section 3.1.3 and the hierarchy is described in appendix section 4.
In Figure 13, the number of occurrences of each class for all documents is described:
while the number of observations for each class in each document is analyzed in the appendix section 21. As it can be noted, there is also some imbalance in our classes. Therefore, we test the models again using a balanced method with our model but also the sentences generated from the ChatGPT, where the number of observations for each class is described in the graph 14.
Based on that, we get the results as shown in Table 7.

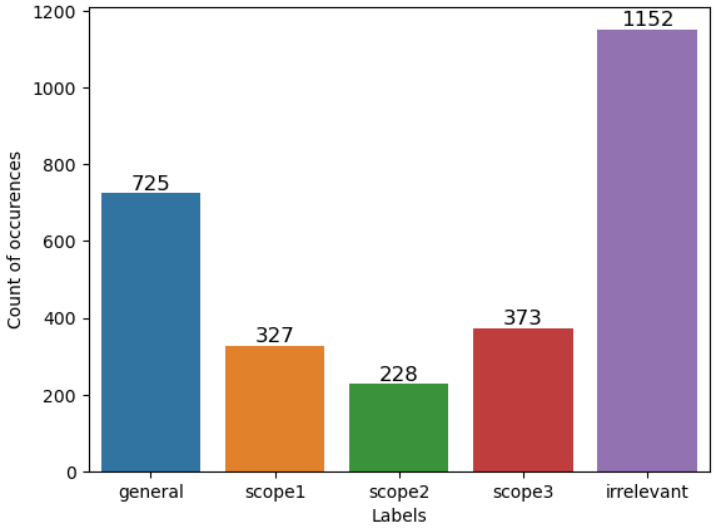


Figure 13: Number of occurrences for each class for all documents.

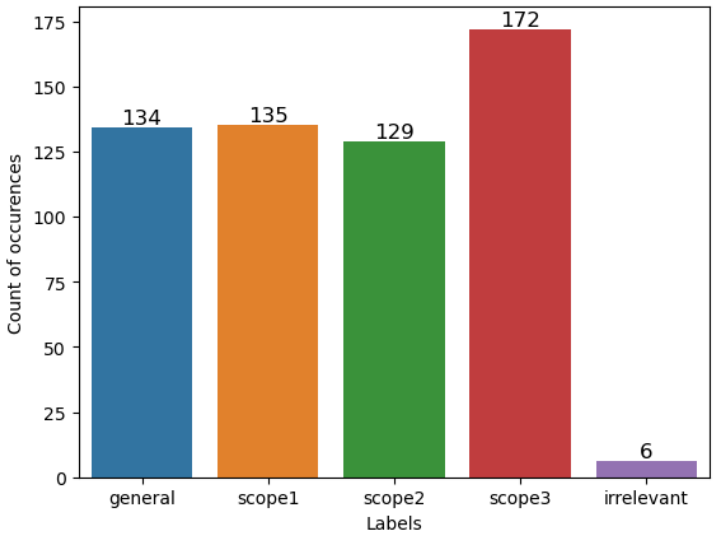


Figure 14: Number of occurrences for each class in the ChatGPT sentences.

5.3.2 Create a new classification for the Scopes

For the second method, we do not make a hierarchical classification but we design two different models. The first classifier classifies the sentences into the three main categories while the second one generates the second label, meaning the categorization to Scopes 1, 2 and 3. For this method, we categorise the sentences that ESG BERT has set the label GHG_Emissions to relevant and irrelevant and after this classification, we categorise them into the tree Scopes. Again we test all the models for the binary classification into relevant/irrelevant as in the previous steps and we get the results as described in Table 8.

Next, we get only the relevant sentences and we classify them into Scopes 1, 2 and 3, creating a multi-label text classification. For this test, we do not use the general label but we classify them with all three other labels. The design for this method is depicted in the appendix section

Model	Balance Method	Precision	Recall	F1-score	Accuracy
RF	None	0.77	0.79	0.76	0.70
RF	SMOTE	0.80	0.80	0.77	0.97
RF	Near Miss v2	0.60	0.55	0.54	0.60
RF	ChatGPT	0.79	0.77	0.76	0.96
MNB	None	0.54	0.63	0.56	0.63
MNB	SMOTE	0.70	0.55	0.55	0.55
MNB	Near Miss v2	0.62	0.40	0.39	0.40
MNB	ChatGPT	0.62	0.64	0.59	0.64
SVC	None	0.74	0.76	0.73	0.76
SVC	SMOTE	0.76	0.75	0.75	0.98
SVC	Near Miss v2	0.63	0.52	0.53	0.55
SVC	ChatGPT	0.77	0.75	0.73	0.75
BERT	None	0.83	0.83	0.81	0.83
BERT	**ChatGPT**	**0.90**	**0.88**	**0.87**	**0.87**

Table 7: Experiments of different models for multi-class classification for the annotation in Scopes with different methods for balancing the dataset and their precision, recall, F1-score and accuracy results.

4.2.3,6. We also include the accuracy of the model as a metric for the comparison of these two methods. It is worth noting that for multi-label classification we do not use a balancing method, as these methods are not directly applicable to these types of models, but we test the models with and without the sentences from ChatGPT. The results for the final step are shown in Table 9.

Model	Balance Method	Precision	Recall	F1-score	Accuracy
RF	None	0.91	0.90	0.90	0.90
RF	SMOTE	0.90	0.90	0.90	0.90
RF	Near Miss v2	0.90	0.89	0.89	0.89
RF	ChatGPT	0.90	0.88	0.88	0.88
MNB	None	0.82	0.76	0.75	0.79
MNB	SMOTE	0.84	0.78	0.77	0.62
MNB	Near Miss v2	0.82	0.76	0.75	0.61
MNB	ChatGPT	0.81	0.69	0.66	0.52
SVC	None	0.88	0.86	0.86	0.86
SVC	SMOTE	0.88	0.86	0.86	0.86
SVC	Near Miss v2	0.88	0.86	0.86	0.86
SVC	ChatGPT	0.88	0.86	0.86	0.86
BERT	**None**	**0.94**	**0.94**	**0.93**	**0.93**
BERT	**ChatGPT**	**0.94**	**0.94**	**0.93**	**0.94**

Table 8: Experiments of different models for binary classification for the Relevant and Irrelevant task for the second label in our task with different methods for balancing the dataset and their precision, recall, F1-score and accuracy results.

Model	Balance Method	Precision	Recall	F1-score	Accuracy
RF	None	0.80	**0.99**	0.88	0.63
RF	ChatGPT	0.85	0.97	0.90	**0.69**
MNB	None	0.80	0.99	0.88	0.63
MNB	ChatGPT	0.85	0.97	0.90	**0.69**
SVC	None	0.80	0.99	0.88	0.63
SVC	ChatGPT	0.85	0.97	**0.90**	**0.69**
BERT	None	**0.91**	0.76	0.82	0.45
BERT	ChatGPT	**0.91**	0.76	0.82	0.45

Table 9: Experiments of different models for multi-label classification for the second step with different methods for balancing the dataset and their precision, recall, F1-score and accuracy results.

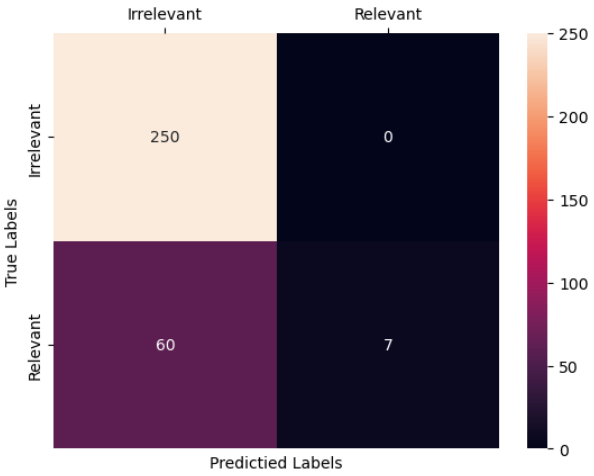


Figure 15: Confusion Matrix for Multinomial Naive Bayes using TF-IDF

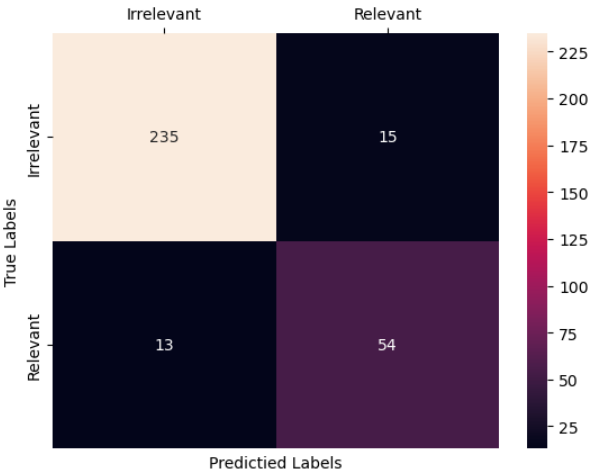


Figure 16: Confusion Matrix for Multinomial Naive Bayes without using TF-IDF

5.4 Information extraction from the tables

Regarding the information extraction from tables, as we excluded the sentences which are in a table in each document, we do not have this information if we implement only the sentence classification. Thus, we take the images that are extracted from the table detection model as referenced in 3.1.1, and we utilize the PyMuPDF algorithm, which extracts the text from the image by converting the image into pdf again. Next, we annotate the tables as relevant and irrelevant to the task and therefore the task can be referred to as binary classification. Again, we use the same models that have been used in the hierarchical classification, without using an oversampling or under-sampling method. As the amount of data is really low, only 317 tables, we do not use cross-validation per dataset, but cross-validation with 10 files split. The results of this task are shown in Table 10:

Model	**Use TFIDF**	**Precision**	**Recall**	**F1-score**
RF	True	0.95	0.35	0.55
RF	False	0.80	0.33	0.48
MNB	**True**	**0.50**	**0.11**	**0.17**
MNB	**False**	**0.86**	**0.81**	**0.80**
SVC	True	0.90	0.32	0.46
SVC	False	0.70	0.20	0.29
BERT	False	0.66	0.60	0.62

Table 10: Experiments of different models for table extraction

It is worth noting, the big difference that models present if we use TF-IDF or not as can be seen in Figures 15 and 16.

6 Discussion

In this section, we will discuss the results of our experiments in the context of the research questions, by summarizing our main findings during our experiments.
Starting with the classification of relevant and irrelevant classes from all sentences in every document we observe that if we do not use any balancing method, the relevant sentences are categorized as irrelevant, while the actually irrelevant sentences are classified correctly, as is seen in Figure 18. On the other hand, if we use the undersampling method, meaning NearMiss, most of the relevant sentences are classified correctly, but the model returns also many irrelevant sentences as relevant, as is depicted in Figure 20. Moreover, using SMOTE as an oversampling method, we do not get any relevant sentences, but we get some irrelevant sentences as relevant, as we observe in Figure 19. In other words, balancing methods cannot handle the huge gap that exists between the two classes, as we get almost 15 thousand irrelevant sentences and only 383 relevant ones. In order to handle this limitation and prepare our model for the next step, we use a fine-tuned BERT model in ESG chapters, named ESG BERT, and we select only the GHG_Emissions label. If we observe the results for documents in Figure 11, we observe that most of the relevant sentences return them correctly while the categorization of irrelevant sentences is significantly better than the SVC model as ESG BERT returns a much lower number of wrongly relevant sentences.
The next step is the categorization of Carbon Footprint, Reduction of Carbon Emissions and Target. If we use flattened classification, meaning that we try to classify them, using all the sentences of a document and classify them into these categories or Irrelevant, we get similar results as in the previous step. Again, the models classified all the sentences wrongly as irrelevant when not using any balancing method, as it is shown in Figure 22, while if we do, the results are slightly better but again we need to deal with the same limitations as the first step. On the other hand, if we follow the hierarchical classification approach and try to classify only the sentences from the ESG_Emissions class, the results are remarkably better for every model that we have tested. This can be easily seen in Table 23, as the precision, recall and F1-score are on average 80 % for every model. This could be happening, due to the irrelevant words subtracted when ESG BERT is implemented, as the most relevant words that can be seen in Figure 10 are irrelevant to our subject, while the words that are depicted in Figure 12, after the ESG-BERT implementation, are more related. The BERT model returns the best results for this classification as the precision is 86%, the recall 85% and the F1-score is 85%. We observe that for every document tested, the results are good, and it is noteworthy that in certain cases the model achieves remarkable results reaching 100% in terms of precision and recall per class, as it can be seen in Table 8. The second best model is the Random Forest Classifier without a balancing method utilized, which has all its metrics around 69%. Trying to make the results better, we implement a different method to balance the data, meaning that we add some sentences generated from ChatGPT by using the question: " Can you give me 100 sentences for carbon emissions related to amounts, reduction or targets in business?" and then rephrase them in order to take unique words related to our task. If we observe the results of the BERT model after the enrichment of the training set, the precision score gets higher by 4% reaching 90%. The same happens with the F1-score, which finally is 89% while the recall has a slightly lower increase, reaching 87%. The most important increase to the metrics, after adding the sentences from ChatGPT is in the classification with the Multinomial Naive Bayes model, where precision, recall and F1-score are improved by 6, 7 and 9% respectively.
The final step in our research is to annotate the sentences by Scope. In this step, we skip the classification containing all the sentences from the documents and instead use only the sentences that are classified from ESG-BERT. Two methods are used for this annotation. The first one is to continue the hierarchy from the previous step, meaning that we have the primary labels

and we classify them into Irrelevant, General, in which the sentences that are contained have information for all Scopes, Scope 1, Scope 2 and Scope 3. In our second method, we start by classifying the sentences into relevant and irrelevant. Next, we categorise them into scopes 1, 2 and 3 using multi-label classification instead of just one label. In the former method, the Multinomial Naive Bayes model with the enrichment of the training set of the ChatGPT sentences seems to not respond to the categorisation as its precision, recall and F1-score are 62%,64% and 59% respectively. On the other hand, SVC and Random Forest Classifier exhibit better results, as their metrics, including the sentences from ChatGPT, reach on average 75% and 77.5%. The best model for this classification again is the BERT model, which presents significant results for this categorisation, as its precision, recall and F1-score are around 88% using the same method as the other models. If we observe the results for all documents as a test in Tables 8,8, we see that the model cannot achieve the same scores as in the previous step, but again in some cases, the predictions for each class are excellent. In the latter method, the binary classification has remarkable results in spite of the model that we use, as all of them perform well. For this task, model comparison is done by comparing the recall of the models, as it is an important metric because high recall means that the model is good at capturing a large portion of the true positive instances in the dataset and it has a low number of false negatives. Random Forest Classifier's recall without any balancing method or by using SMOTE extends to 90%, while with ChatGPT instances or by using NearMiss the recall is slightly lower. As for SVC the recall does not change despite the balance method that we use and it is stable at 86%. If we observe the results when we use the Multinomial Naive Bayes, we notice that it is again the worst model of our selection. Regarding the BERT model, the recall remains constant at 94% even if we include the sentences from ChatGPT. Continuing to the next step, we compare the models, not only with the three measures that we discussed in the previous steps but we also include the accuracy. Overall, the aforementioned models present good precision, recall and F1-score, however, Random Forest Classifier outperforms all of them, reaching its metrics in really high scores. Although, the accuracy score is not so high, as is shown in Table 9. This shows that the model is performing very well in terms of correctly identifying individual labels and reproducing all relevant instances of each label, though it struggles with correctly classifying sentences with multiple labels. In other words, low accuracy suggests that the model is making mistakes in predicting the combination of labels correctly for each sentence.

Comparing the two methods for the labelling of the Scopes, we observe that even though the latter method presents better results, it struggles to find all the labels for each sentence while the former method has lower recall and F1 -score but the accuracy of the model is 87%, that points to better labelling since the accuracy is almost the same to other metrics which leads us to conclude that overall the model can predict the labels better.

As for the Information Extraction from the tables, we test all the models as previously. The main difference is that we test the models with and without TF-IDF. The results are really interesting, as we saw that without TF-IDF the Multinomial Naive Bayes model presents a significant difference in the classification report, when run with TF-IDF only 7 from the relevant tables are categorized correctly and 60 are not while not running it with TF-IDF then 54 are classified correctly and 13 are not. This may be done due to the fact that a table includes more numerical data than text. Therefore, TF-IDF may not work in this case as it cannot give relations to the tokens. As in the tabular data, more numbers than words are included it seems that TF-IDF cannot recognize the most frequent words in the documents. Therefore it does not return efficient numbers as it cannot recognise important words in the documents. On the other hand, if we observe the irrelevant tables, in the former test none of them were classified wrongly, while in the latter 13 of them are found as relevant. Regarding the other models, SVC and Random Forest Classifier without TF-IDF present similar results in their metrics. Both of them have low recall and F1-score even though their precision score is high. Now if we test them using

the TFIDFTransofrmer, the metrics for both models are a bit better but again the models fail to outperform the Multinomial Naive Bayes model. Concerning the BERT model, we perceive that it is better than Random Forest Classifier and SVC, because of the fact that its metrics are more balanced but again it cannot achieve better results than the Multinomial Naive Bayes model, as its precision, recall and F1-score are 66%, 60% and 62% respectively, while the best model achieves 86%, 81% and 80% in the same measures.

7 Conclusion

This research was motivated by the necessity to extract information about carbon emissions in the industry field. Given annual reports, we need to find a way to extract this information emphasizing to be as effective as possible. We use eleven sustainability documents and we annotate them. In addition, we experiment with four different models, using some methods to imbalance the data and we try to enrich it using the ChatGPT, by generating sentences related to the task. Moreover, we utilize flattened multi-class classification, hierarchical multi-class and multi-label classification. Consequently, after performing all experiments and discussing the results, we answer the following research question:

To what extent is it possible to extract information regarding carbon emissions from businesses' annual reports?
This question can be answered by solving the following sub-questions:

What kind of information can we extract from an annual report of a company relating to carbon emissions?

We observed that we can find a way to extract sentences related to carbon emissions despite the fact that in most cases their number was really low in each document, as presented in Figure 7, by using the ESG-BERT model and subtracting other categories. Also, there was a way to classify them into the three main categories **Carbon Emissions**, **Reduction of carbon Emissions**, and **Target** but also into the three Scopes that the European Union suggests. Moreover, we found that table extraction was possible, by editing the PDF files and using pre-trained models for table detection in the PDF files. After that, the comparative tables were able to be extracted by using a binary classification model.

How can we produce quality data for this purpose?

This question regards the task of extracting the related sentences. We proposed a way to annotate the data by using the pyMuPDF library to transform the PDF file into text in order to be tokenized into sentences. Next, after the implementation of NLTK and spaCy library, we found a way to make the tokenization better as we observed cases where the tokenization was not correct. We fixed the problem where a full stop did not exist at the end of a sentence when there was an upper letter but not a full stop or the occurrence to change a line and finally, the case that there were more spaces than one but again not a full stop. After utilizing this algorithm we implemented again the NLTK library and we tokenized the text into sentences. In addition, we implemented the ESG-BERT and annotated the sentences with one label referring to the three main labels and one referring to the Scope label.

Which models are the most effective for this problem?

We propose the hierarchal classification model, starting from the implementation of the ESG-BERT model. Next, we implement the BERT model trained with the ChatGPT sentences in

order to find the main label for the sentence. The following step is to categorise the data into Scopes, using an additional label, General, that referred to sentences that include information for two or more Scopes. Again, we used the BERT model trained with the enrichment of ChatGPT sentences. The pipeline for the final model can be seen in Figure 17.

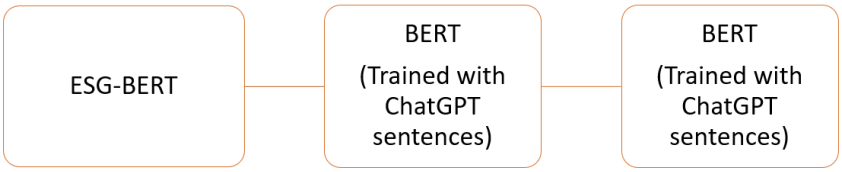


Figure 17: Pipeline for the proposed hierarchical model, describing the model that is used for each step.

As for the table extraction, we propose the Multinomial Naive Bayes model, without using the TF-IDF transformer in our data but only CountVectorizer.

By answering all these sub-questions we can extract the data structured and labelled. A sample of the model's output is depicted in the Table 11

Sentence	Output of ESG-BERT	Output of Cabron Footprint Classifier	Output of Scopes Classifier
the boreas is our first ship suitable for operating on green methanol, an e-fuel with very low carbon emissions	GHG_Emissions	Reduction Of Carbon Emissions	General
the 900 mw greater changhua 1 & 2a offshore wind farm is the first cable project in taiwan for company	GHG_Emissions	Irrelevant	Irrelevant
at the same time, our clients expect us to work on reducing our carbon footprint	GHG_Emissions	**Irrelevant**	General
in addition, we are committed to using soil that is native to the area, executing our work without generating emissions and using materials in an efficient way	GHG_Emissions	Reduction Of Carbon Emissions	General
meanwhile, the majority of company's total emissions fall under scope 3	GHG_Emissions	Carbon Footprint	Scope 3
with natural gas reserves of 2.8 trillion cubic metres, mozambique is poised to play a key role in the emerging african lng sector	GHG_Emissions	**Irrelevant**	**Scope 1**
we encourage the registration of the impact on greenhouse gas emissions, air pollution, use of natural resources and the use of plastics	GHG_Emissions	Irrelevant	Irrelevant
the vessel will have a large battery pack, a shore supply connection and a state-of-the-art energy management system, meaning reduced co2, nox and sox (carbon, nitrogen and sulphur oxides) emissions	GHG_Emissions	Target	General

Table 11: A sample of the resulting output of the model, giving one document as an example. With bold words, we annotate the wrongly labelled sentences from the model.

This project can be used by companies in order to summarize the information regarding carbon emissions from annual reports. This can be useful for consulting other companies or creating predicted models. Moreover can be used in order to compare companies but also for governments to control countries' emissions.

The repository for PDF transformation into sentences can be found in https://github.com/MariannaKf/pdf_structured_data

7.1 Future work

For future work, we can try to extract also the related images from the PDF. Another idea is the extraction and the categorisation of other chapters like Air-quality or Energy management.

Also, the tables that are extracted can be edited and transformed into dataframes to analyze and visualize the results. Moreover, we could test more pre-trained models for text classification like ELECTRA or enrich data with more annotation of annual reports. Also, we could use methods to extend the training dataset, like web scraping or more usage of ChatGPT, such as testing the possibility to annotate the annual reports automatically instead of manually.

References

[1] Paris Agreement. https://unfccc.int/process-and-meetings/the-paris-agreement/the-paris-agreement?gclid=Cj0KCQjwwfiaBhC7ARIsAGvcPe4rITe6KCPj5dvSHQBjXn9yQ_kW5vmdp2sgIHneJwUhrqyvCy-bGYQaAkyLEALw_wcB. Date accessed: 30-10-2022.

[2] ESG protocol. ://www.esgportal.eu/carbon-footprint/. Date accessed: 30-10-2022.

[3] Hannah Bast and Claudius Korzen. A benchmark and evaluation for text extraction from pdf. In *2017 ACM/IEEE joint conference on digital libraries (JCDL)*, pages 1–10. IEEE, 2017.

[4] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47, 2002.

[5] Jeremy J Eberhardt. Bayesian spam detection. *Scholarly Horizons: University of Minnesota, Morris Undergraduate Journal*, 2(1):2, 2015.

[6] Alec Go, Lei Huang, Richa Bhayani, et al. Twitter sentiment analysis. *Entropy*, 17:252, 2009.

[7] Carlos N Silla and Alex A Freitas. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 22:31–72, 2011.

[8] Daphne Koller and Mehran Sahami. Hierarchically classifying documents using very few words. In *ICML*, volume 97, pages 170–178, 1997.

[9] Miguel E Ruiz and Padmini Srinivasan. Hierarchical text categorization using neural networks. *Information retrieval*, 5:87–118, 2002.

[10] Wei-Cheng Chang, Hsiang-Fu Yu, Kai Zhong, Yiming Yang, and Inderjit S Dhillon. Taming pretrained transformers for extreme multi-label text classification. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 3163–3171, 2020.

[11] Eduardo Tieppo, Roger Robson dos Santos, Jean Paul Barddal, and Júlio Cesar Nievola. Hierarchical classification of data streams: a systematic literature review. *Artificial Intelligence Review*, pages 1–40, 2022.

[12] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[13] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[14] Random Forest Classifier. https://towardsdatascience.com/understanding-random-forest-58381e0602d2. Date accessed: 17-04-2023.

[15] Hinge loss. https://en.wikipedia.org/wiki/Hinge_loss. Date accessed: 17-04-2023.

[16] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

[17] Geoffrey I. Webb. *Naïve Bayes*, pages 713–714. Springer US, Boston, MA, 2010.

[18] BERT. https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270. Date accessed: 17-04-2023.

[19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[20] Azka Gilani, Shah Rukh Qasim, Imran Malik, and Faisal Shafait. Table detection using deep learning. In *2017 14th IAPR international conference on document analysis and recognition (ICDAR)*, volume 1, pages 771–776. IEEE, 2017.

[21] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 213–229. Springer, 2020.

[22] Brandon Smock, Rohith Pesala, and Robin Abraham. Pubtables-1m: Towards comprehensive table extraction from unstructured documents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4634–4642, 2022.

[23] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[24] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[25] Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284, 2009.

[26] Anil Vinjamur, Sumali Conlon, Susan Lukose, Tim McCready, and Jason Hale. Automatic extraction and generation of xml documents from financial reports. 2005.

[27] Valter Crescenzi and Giansalvatore Mecca. Automatic information extraction from large websites. *Journal of the ACM (JACM)*, 51(5):731–779, 2004.

[28] Thomas Bayer, Ulrich Bohnacker, and Ingrid Renz. Information extraction from paper documents. In *Handbook of Character Recognition and Document Image Analysis*, pages 653–677. World Scientific, 1997.

[29] Pan Ding, Liang Zhuoqian, and Deng Yuan. Textual information extraction model of financial reports. In *Proceedings of the 2019 7th International Conference on Information Technology: IoT and Smart City*, pages 404–408, 2019.

[30] Tomoki Ito, Hiroki Sakaji, and Kiyoshi Izumi. Segment information extraction from financial annual reports using neural network. In *Annual Conference of the Japanese Society for Artificial Intelligence*, pages 215–226. Springer, 2019.

[31] Filip Graliński, Tomasz Stanisławek, Anna Wróblewska, Dawid Lipiński, Agnieszka Kaliska, Paulina Rosalska, Bartosz Topolski, and Przemysław Biecek. Kleister: A novel task for information extraction involving long documents with complex layout. *arXiv preprint arXiv:2003.02356*, 2020.

[32] Ramzan Talib, Muhammad Kashif Hanif, Shaeela Ayesha, and Fakeeha Fatima. Text mining: techniques, applications and issues. *International Journal of Advanced Computer Science and Applications*, 7(11), 2016.

[33] Amir Mohammad Shahi, Biju Issac, and Jashua Rajesh Modapothala. Analysis of supervised text classification algorithms on corporate sustainability reports. In *Proceedings of 2011 International Conference on Computer Science and Network Technology*, volume 1, pages 96–100. IEEE, 2011.

[34] Jing Fang, Liangcai Gao, Kun Bai, Ruiheng Qiu, Xin Tao, and Zhi Tang. A table detection method for multipage pdf documents via visual seperators and tabular structures. In *2011 International Conference on Document Analysis and Recognition*, pages 779–783. IEEE, 2011.

[35] Amir Riad, Christian Sporer, Syed Saqib Bukhari, and Andreas Dengel. Classification and information extraction for complex and nested tabular structures in images. In *2017 14th IAPR international conference on document analysis and recognition (ICDAR)*, volume 1, pages 1156–1161. IEEE, 2017.

[36] Apple Environmental Progress Report. https://www.apple.com/environment/pdf/Apple_Environmental_Progress_Report_2022.pdf. Date accessed: 21-06-2023.

[37] Microsoft 2021 Environmental Sustainability Report. https://query.prod.cms.rt.microsoft.com/cms/api/am/binary/RE4RwfV. Date accessed: 21-06-2023.

[38] The Coca-Cola Company 2021 Business ESG Report. https://www.coca-colacompany.com/content/dam/journey/us/en/reports/coca-cola-business-environmental-social-governance-report-2021.pdf. Date accessed: 21-06-2023.

[39] FY21 NIKE Inc. Impact Report. https://about.nike.com/en/newsroom/reports/fy21-nike-inc-impact-report-2. Date accessed: 21-06-2023.

[40] IKEA Climate Report FY21. https://gbl-sc9u2-prd-cdn.azureedge.net/-/media/aboutikea/newsroom/publications/documents/ikea-climate-report-fy21.pdf?rev=1f887c6a9dc948f18e20cd378983a1a8. Date accessed: 21-06-2023.

[41] Samsung Electronics Sustainability Report 2021. https://image-us.samsung.com/SamsungUS/home/pdf/Samsung-Electronics-Sustainability-Report-2021.pdf. Date accessed: 21-06-2023.

[42] KLM, Royal Dutch Airlines, Annual Report 2021. https://img.static-kl.com/m/7f18a4405ec39c57/original/KLM-2021-Annual-Report.pdf. Date accessed: 21-06-2023.

[43] Annual Report Royal Van Oord 2021. https://vanoordjaarverslagcms.appstudio.nl//upload/131/pdfs/file/Van%20Oord%20Annual%20Report%202021.pdf. Date accessed: 21-06-2023.

[44] Amazon's 2021 Sustainability Report. https://sustainability.aboutamazon.com/2021-sustainability-report.pdf. Date accessed: 21-06-2023.

[45] Meta - 2021 Sustainability Report . https://sustainability.fb.com/wp-content/uploads/2022/06/Meta-2021-Sustainability-Report.pdf. Date accessed: 21-06-2023.

[46] Mercedes-Benz Sustainability Report 2021 . https://group.mercedes-benz.com/documents/sustainability/other/mercedes-benz-sustainability-report-2021.pdf. Date accessed: 21-06-2023.

[47] GRG protocol. https://ghgprotocol.org/. Date accessed: 19-10-2022.

[48] GRI standards for reporting. https://www.globalreporting.org/how-to-use-the-gri-standards/gri-standards-english-language/. Date accessed: 19-10-2022.

8 Appendix

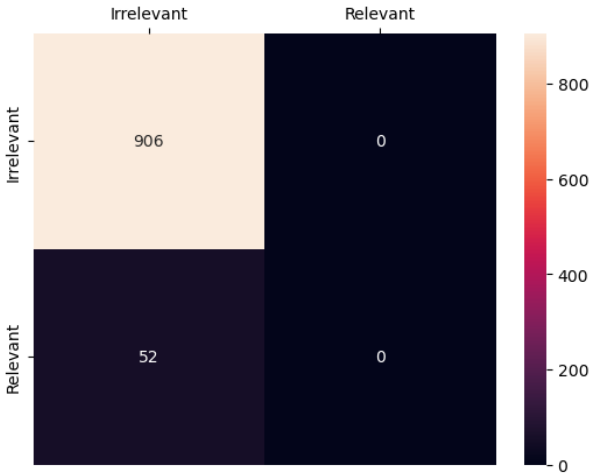


Figure 18: Confusion Matrix for Irrelevant/Relevant categorization using Multinomial Naive Bays without any balancing method.

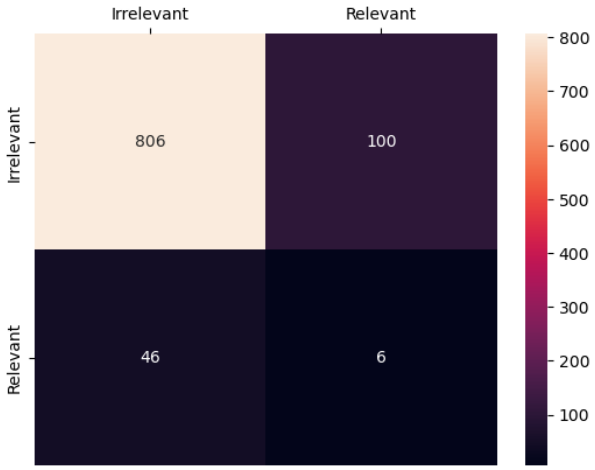


Figure 19: Confusion Matrix for Irrelevant/Relevant categorization using Multinomial Naive Bays with SMOTE as balancing method.

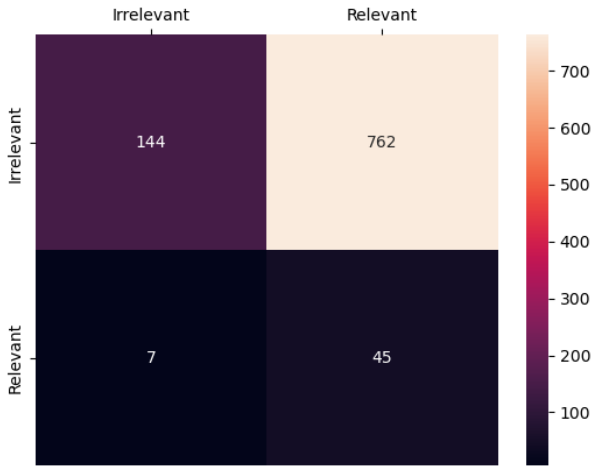


Figure 20: Confusion Matrix for Irrelevant/Relevant categorization using Support Vector Classifier with Near Miss version 1 as balancing method.

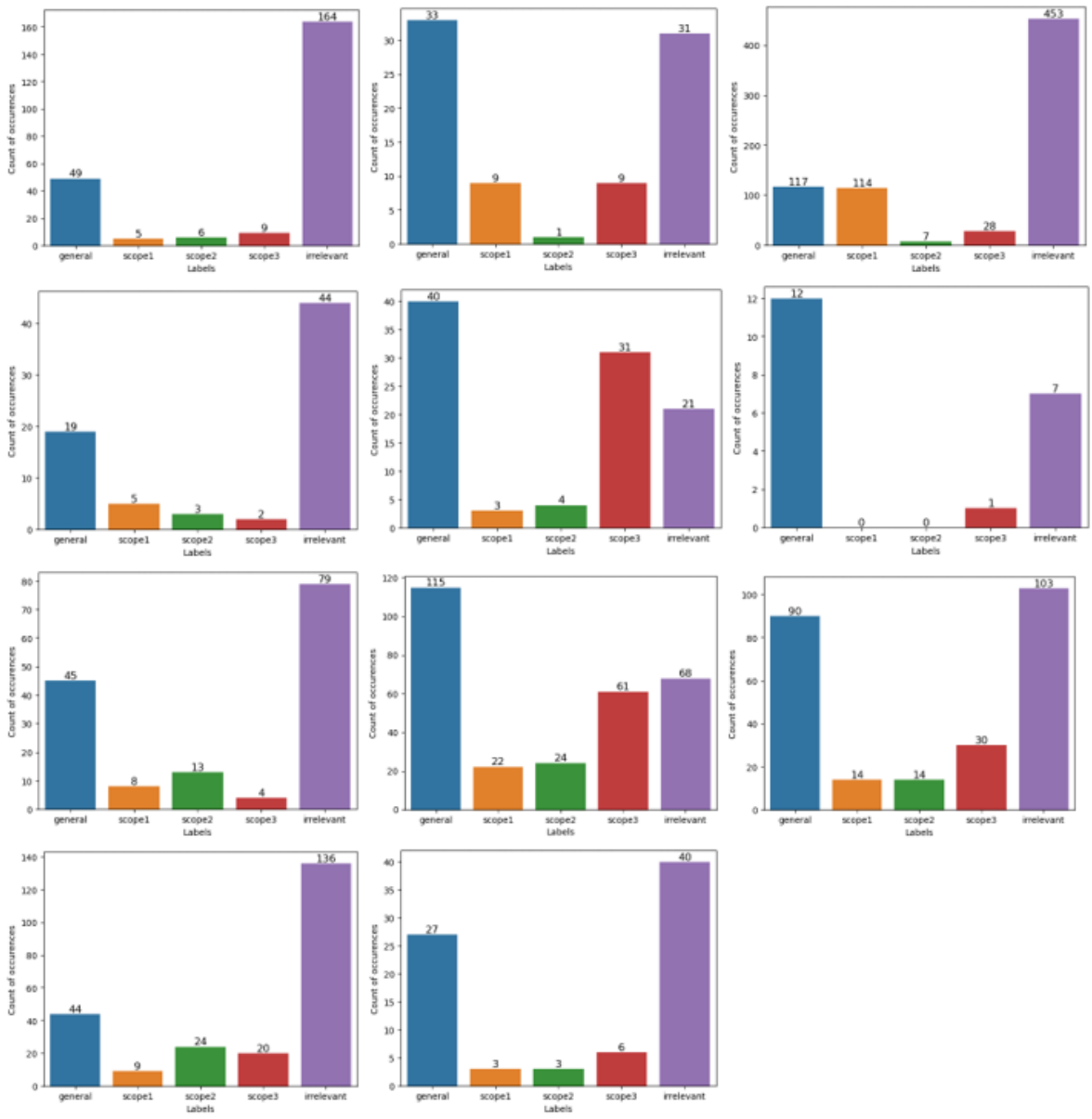


Figure 21: Number of occurrences for every document separately for the final classification in the hierarchical classification.

Categories of ESG BERT
GHG_Emissions
Air_Quality
Energy_Management
Physical_Impacts_Of_Climate_Change
Ecological_Impacts
Product_Design_And_Lifecycle_Management
Business_Model_Resilience
Water_And_Wastewater_Management
Product_Quality_And_Safety
Human_Rights_Community_Relations
Supply_Chain_Management
Director_Removal
Selling_Practices_And_Product
Customer_Welfare
Systemic_Risk_Management
Business_Ethics
Data_Security
Waste_And_Hazardous_Materials_Management
Competitive_Behavior
Critical_Incident_Risk_Management
Employee_Health_And_Safety
GHG_Emissions
Employee_Engagement_Inclusion_And_Diversity
Management_Of_Legal_Regulatory_Framework

Table 12: Output categories of The ESG-BERT model

Samples of Chat GPT model
Our company's Scope 1 emissions come from direct sources such as the burning of fossil fuels in our own facilities
Our Scope 2 emissions are indirect emissions from the consumption of purchased electricity, heat or steam.
Scope 3 carbon emissions from supply chain logistics are a significant contributor to Company S's carbon footprint.
Company B emitted 100,000 metric tons of CO2e in total carbon emissions last year.
We have implemented measures to reduce our Scope 1 emissions by switching to renewable energy sources for our production processes.
We have signed contracts with renewable energy suppliers to reduce our Scope 2 emissions and increase the proportion of green energy we use.
As part of our sustainability strategy, we are encouraging our suppliers to adopt more environmentally friendly practices to help reduce our Scope 3 emissions.
We are using life cycle assessment (LCA) to evaluate the carbon footprint of our products and identify ways to reduce emissions throughout the product's lifecycle.
The majority of our Scope 1 emissions come from our fleet of vehicles, which we plan to replace with electric models over the next five years.
Renewable energy targets: Companies might set targets to increase the proportion of renewable energy they use, such as a goal to source 100% of their electricity from renewable sources by 2030.
Companies can also set targets for their supply chain emissions, such as a goal to reduce the carbon emissions associated with their suppliers by 25% by 2025.
Many companies are aiming to achieve net-zero carbon emissions, meaning they will either eliminate or offset all of the greenhouse gases they produce.

Table 13: Samples of sentences generated from ChatGPT

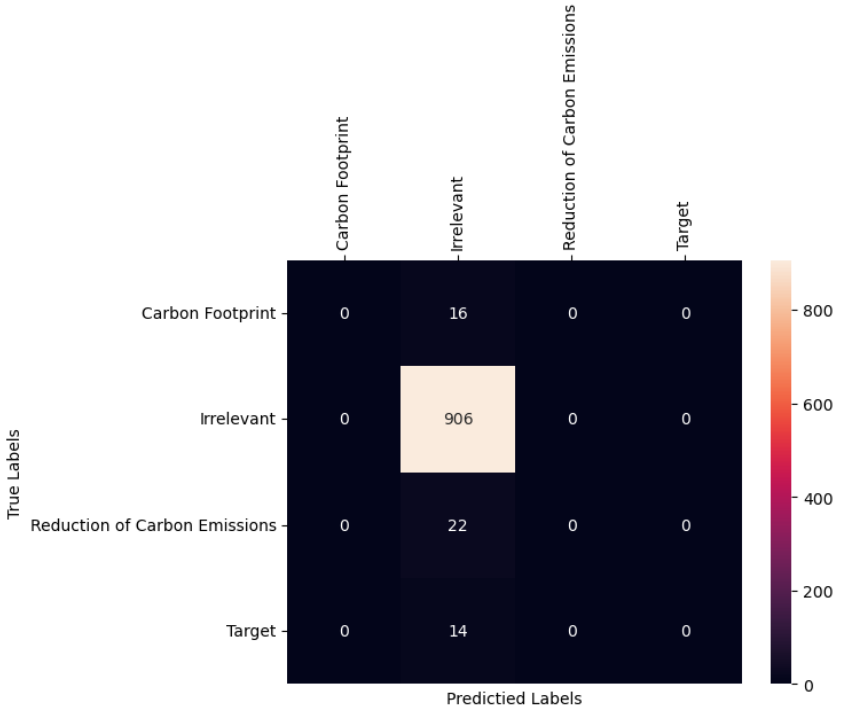


Figure 22: Confusion Matrix for categorization in Carbon Footprint, Reduction of Carbon Emissions and Target without using ESG-BERT model for sentences excluding, using BERT model.

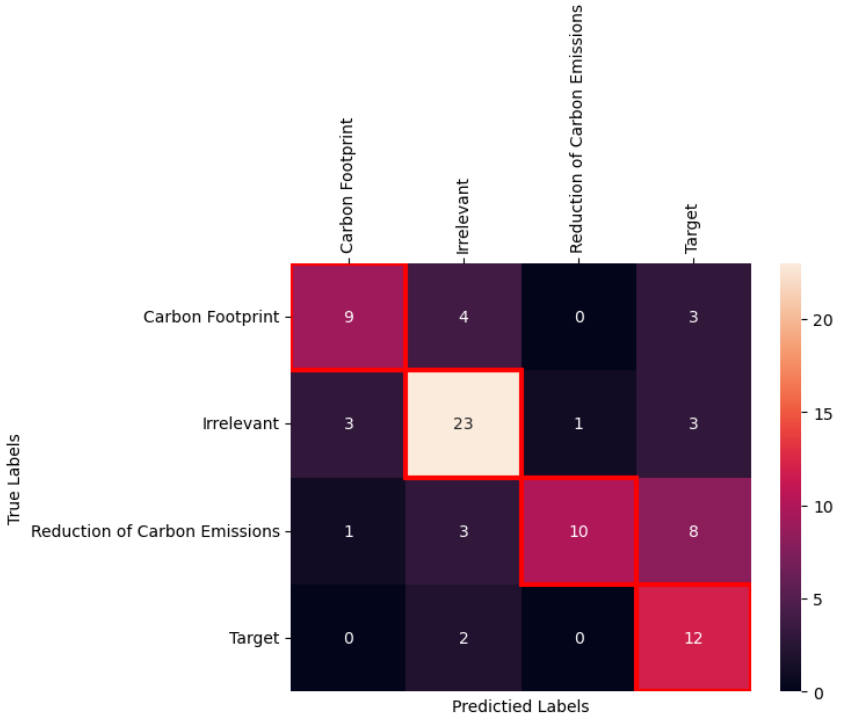


Figure 23: Confusion Matrix for categorization in Carbon Footprint, Reduction of Carbon Emissions and Target using ESG-BERT model for sentences excluding, using BERT model and enrich the data with ChatGPT sentences.

Class		Classification Report		
		Precision	**Recall**	**F1-score**
Document 1	Irrelevant	0.52	0.73	0.60
	Carbon Footprint	0.81	0.60	0.69
	Reduction Of Carbon Emissions	0.50	0.92	0.65
	Target	0.86	0.45	0.60
Document 2	Irrelevant	0.91	0.89	0.90
	Carbon Footprint	0.74	0.73	0.74
	Reduction Of Carbon Emissions	0.70	0.79	0.75
	Target	0.49	0.63	0.55
Document 3	Irrelevant	0.95	0.98	0.97
	Carbon Footprint	0.93	0.93	0.93
	Reduction Of Carbon Emissions	1.00	0.77	0.87
	Target	0.60	1.00	0.73
Document 4	Irrelevant	1.00	1.00	1.00
	Carbon Footprint	1.00	1.00	1.00
	Reduction Of Carbon Emissions	1.00	0.83	0.91
	Target	0.80	1.00	0.89
Document 5	Irrelevant	1.00	0.78	0.88
	Carbon Footprint	0.88	1.00	0.93
	Reduction Of Carbon Emissions	1.00	0.93	0.96
	Target	0.89	1.00	0.94
Document 6	Irrelevant	0.97	0.94	0.96
	Carbon Footprint	0.97	0.95	0.96
	Reduction Of Carbon Emissions	0.86	0.97	0.91
	Target	0.94	0.97	0.96
Document 7	Irrelevant	0.99	0.99	0.99
	Carbon Footprint	0.96	0.99	0.97
	Reduction Of Carbon Emissions	1.00	0.79	0.88
	Target	0.91	1.00	0.95
Document 8	Irrelevant	0.96	1.00	0.98
	Carbon Footprint	1.00	0.95	0.97
	Reduction Of Carbon Emissions	0.89	1.00	0.94
	Target	1.00	0.93	0.96
Document 7	Irrelevant	0.99	0.99	0.99
	Carbon Footprint	0.96	0.99	0.97
	Reduction Of Carbon Emissions	1.00	0.79	0.88
	Target	0.91	1.00	0.95
Document 8	Irrelevant	0.96	1.00	0.98
	Carbon Footprint	1.00	0.95	0.97
	Reduction Of Carbon Emissions	0.89	1.00	0.94
	Target	1.00	0.93	0.96
Document 9	Irrelevant	1.00	1.00	1.00
	Carbon Footprint	0.87	1.00	0.93
	Reduction Of Carbon Emissions	1.00	1.00	1.00
	Target	1.00	0.83	0.91
Document 10	Irrelevant	0.97	1.00	0.98
	Carbon Footprint	0.96	0.99	0.97
	Reduction Of Carbon Emissions	0.92	0.92	0.92
	Target	0.94	0.97	0.96
Document 11	Irrelevant	0.99	0.99	0.99
	Carbon Footprint	1.00	0.97	0.98
	Reduction Of Carbon Emissions	1.00	1.00	1.00
	Target	0.96	0.96	0.96

Table 14: Classification Reports for BERT model with sentences from ChatGPT for cross-validation per document for classes Carbon Footprint, Reduction Of Carbon Emissions, Target.

Class		Classification Report		
		Precision	**Recall**	**F1-score**
Document 1	Irrelevant	0.52	0.94	0.67
	General	0.91	0.52	0.66
	Scope 1	0.11	1.00	0.20
	Scope 2	1.00	0.50	0.67
	Scope 3	0.33	0.60	0.43
Document 2	Irrelevant	0.95	0.82	0.88
	General	0.69	0.77	0.73
	Scope 1	0.44	0.766	0.56
	Scope 2	0.86	0.70	0.79
	Scope 3	0.49	0.61	0.54
Document 3	Irrelevant	0.94	0.90	0.92
	General	0.80	0.96	0.87
	Scope 1	1.00	0.93	0.96
	Scope 2	0.42	0.79	0.55
	Scope 3	0.89	0.30	0.45
Document 4	Irrelevant	0.98	1.00	0.99
	General	0.89	0.94	0.92
	Scope 1	0.20	1.00	0.33
	Scope 2	1.00	0.76	0.86
	Scope 3	1.00	0.30	0.44
Document 5	Irrelevant	0.97	0.63	0.77
	General	0.76	0.89	0.82
	Scope 1	0.86	0.90	0.88
	Scope 2	0.67	0.80	0.73
	Scope 3	0.74	0.96	0.83
Document 6	Irrelevant	0.95	0.88	0.92
	General	0.92	0.94	0.93
	Scope 1	0.71	1.00	0.83
	Scope 2	0.57	0.73	0.64
	Scope 3	0.93	0.90	0.92
Document 7	Irrelevant	0.98	0.98	0.98
	General	0.98	0.96	0.97
	Scope 1	1.00	1.00	1.00
	Scope 2	1.00	1.00	1.00
	Scope 3	0.95	0.95	0.95
Document 8	Irrelevant	0.73	0.97	0.83
	General	0.93	0.58	0.72
	Scope 1	0.25	0.33	0.29
	Scope 2	0.50	0.50	0.50
	Scope 3	1.00	0.45	0.62

Table 15: Classification Reports for BERT model with sentences from ChatGPT for cross-validation per document for classes Irrelevant, General, Scope 1, Scope 2, Scope 3 part 1.

Class		Classification Report		
		Precision	**Recall**	**F1-score**
Document 9	Irrelevant	0.93	0.95	0.94
	General	0.89	0.86	0.87
	Scope 1	0.67	0.50	0.57
	Scope 2	0.67	0.40	0.50
	Scope 3	0.60	1.00	0.77
Document 10	Irrelevant	0.87	0.97	0.92
	General	0.78	0.66	0.71
	Scope 1	0.80	0.80	0.80
	Scope 2	0.67	0.57	0.62
	Scope 3	0.89	0.47	0.62
Document 11	Irrelevant	0.99	0.89	0.95
	General	0.80	0.80	0.80
	Scope 1	1.00	1.00	1.00
	Scope 2	1.00	1.00	1.00
	Scope 3	0.90	0.61	0.72

Table 16: Classification Reports for BERT model with sentences from ChatGPT for cross-validation per document for classes Irrelevant, General, Scope 1, Scope 2, Scope 3 part 2.