sustainability

Article

A RoBERTa Approach for Automated Processing of Sustainability Reports

Merih Angin [1], Beyza Taşdemir [2,†], Cenk Arda Yılmaz [2,†], Gökcan Demiralp [2,†], Mert Atay [2], Pelin Angin [2,*] and Gökhan Dikmener [3]

[1] Department of International Relations, Koc University, Istanbul 34450, Turkey
[2] Department of Computer Engineering, Middle East Technical University, Ankara 06800, Turkey
[3] United Nations Development Programme, SDG AI Lab, Istanbul 34381, Turkey
* Correspondence: pangin@ceng.metu.edu.tr
† These authors contributed equally to this work.

Abstract: There is a strong need and demand from the United Nations, public institutions, and the private sector for classifying government publications, policy briefs, academic literature, and corporate social responsibility reports according to their relevance to the Sustainable Development Goals (SDGs). It is well understood that the SDGs play a major role in the strategic objectives of various entities. However, linking projects and activities to the SDGs has not always been straightforward or possible with existing methodologies. Natural language processing (NLP) techniques offer a new avenue to identify linkages for SDGs from text data. This research examines various machine learning approaches optimized for NLP-based text classification tasks for their success in classifying reports according to their relevance to the SDGs. Extensive experiments have been performed with the recently released Open Source SDG (OSDG) Community Dataset, which contains texts with their related SDG label as validated by community volunteers. Results demonstrate that especially fine-tuned RoBERTa achieves very high performance in the attempted task, which is promising for automated processing of large collections of sustainability reports for detection of relevance to SDGs.

Keywords: corporate social responsibility; natural language processing; RoBERTa; sustainable development goals

check for updates

Citation: Angin, M.; Taşdemir, B.; Yılmaz, C.A.; Demiralp, G.; Atay, M.; Angin, P.; Dikmener, G. A RoBERTa Approach for Automated Processing of Sustainability Reports. *Sustainability* **2022**, *14*, 16139. https://doi.org/10.3390/su142316139

Academic Editor: Gwanggil Jeon

Received: 25 September 2022
Accepted: 27 November 2022
Published: 2 December 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Corporate social responsibility (CSR), which can be defined as international self-regulation by private companies that includes a political objective related to both positive and negative environmental and social aspects, has a growing importance in the business world. Despite its contribution to sustainability, under current market conditions, companies seem to be incapable of finding sustainable development solutions on their own. In addition to promoting CSR and eco-efficiency, sustainability requires active participation and cooperation between governments, businesses, and civil society [1].

There is undoubtedly a growing need for effective CSR and sustainability regulations. To meet this need, both voluntary and mandatory initiatives have been launched, which have made sustainability a fundamental part of the corporate agenda. There are numerous voluntary initiatives, some of which are recognized globally, such as the United Nations Global Compact, AA1000, and the Global Reporting Initiative (GRI). These mechanisms have improved business efficiency, but their applicability has been rather limited. There are also Pollutant Release Transfer Register, Carbon Pricing Mechanisms and CSR/Sustainability and Integrated Reporting Requirements as mandatory initiatives, and they have the potential to change the standard approach to CSR. Specific environmental regulations on pollutants and carbon emissions help companies manage their environmental impact and address climate change, while mandatory reporting requirements ensure that a company's CSR activities are known to stakeholders, which facilitate its accountability [2].

The aforementioned developments and initiatives have led investors to start focusing more on corporate sustainability and environmental, social, and governance (ESG) assessments in their investment decisions. As a result, more companies are now voluntarily issuing sustainability reports. Such an increase in sustainability reports is a promising indicator for the future of sustainability; however, we should also note that these reports significantly lack standardization. An analysis on the sustainability reports from the top 20 companies in the S&P 500 [3] shows that the reports vary greatly in terms of length, word count, and number count. It also shows that most of the reported figures are rounded. These findings reveal the need to standardize sustainability reporting and support the ongoing initiatives by regulators that can provide investors with better ESG information.

In this context, one of the most significant initiatives is the Sustainable Development Goals (SDGs) [4] approved by the United Nations General Assembly (UNGA) in 2015, which are intended to be achieved by 2030. SDGs play a key role in facilitating the integration of sustainability to ensure a better and more sustainable future, while responding to the current and future needs of stakeholders and balancing economic, social, and environmental development. SDGs by nature are related to each other, and existing research [5] clearly shows the significant relationships and interlinkages between the 17 SDGs listed below:

1. No poverty;
2. Zero hunger;
3. Good health and well-being;
4. Quality education;
5. Gender equality;
6. Clean water and sanitation;
7. Affordable and clean energy;
8. Decent work and economic growth;
9. Industry, innovation, and infrastructure;
10. Reduced inequalities;
11. Sustainable cities and communities;
12. Responsible consumption and production;
13. Climate action;
14. Life below water;
15. Life on land;
16. Peace, justice, and strong institutions;
17. Partnerships for the goals [4]

SDGs can be broadly categorized under three main topics, namely economy (SDGs 8, 9, 10, and 12), society (SDGs 1, 2, 3, 4, 5, 7, 11, and 16) and environment (SDGs 6, 13, 14, and 15), based on the relevance of their goals, and SDG 17 cross-cuts all of these categories.

Taking into account the notable increase in sustainability reports, approval of 17 SDGs, and the presence of interlinkages between them, utilizing digital platforms and solutions to effectively classify reports according to their relevance to different SDGs is imperative [6]. In this work, we describe a natural language processing (NLP)-based framework that utilizes a fine-tuned RoBERTa model we have built for processing sustainability reports to identify sections relevant to SDGs. Our framework has been built and evaluated using the recently-released OSDG Community Dataset containing textual data on the SDGs, annotated by community volunteers. We have performed extensive experiments to compare the performances of both classical machine learning (ML) models and deep learning (DL) models in binary and multi-class classification tasks demonstrate the superior performance of our model in both tasks. To the best of our knowledge, this is the first work to achieve such high performance in automated classification of sustainability documents from a collection of this size and the first to demonstrate the performance of fine-tuned RoBERTa in this task. We have made the built models readily available for utilization by other researchers and practitioners.

The remainder of this paper is organized as follows: Section 2 summarizes related work in the field of automated text processing for sustainability and other similar domains.

Section 3 provides details of our framework for classification of sustainability reports based on SDGs. Section 4 provides performance analysis of the framework for different ML and DL models, and Section 5 concludes the paper with future work directions.

2. Related Work

With the increasing number of documents and texts created by international organizations and companies everyday, new research efforts have been dedicated to automate the time-consuming process of reading such documents and identifying texts related to target topics such as SDGs.

NLP methods have long been used to automatically process large document collections produced by international organizations. Deniz et al. [7] used NLP to automatically classify sentiments in the large document collection of the International Monetary Fund Executive Board meeting minutes, achieving high accuracy when model training was performed with domain-specific data. Sovrano et al. [8] proposed an ensemble method for multi-label text classification of UNGA Resolutions, combining nondomain-specific deep learning based document similarities with domain-specific term frequency–inverse document frequency (TF-IDF) document similarities. Their proposed method achieved modest performance, but their domain-specific similarity addition improved the baseline without any transfer learning or re-training. Kim and LaFleur [9] proposed a proof-of-concept classifier for analyzing UNGA resolutions adopting the dictionary method and supervised learning. Their proposed classifier achieved 94% test accuracy, and their analysis showed how NLP techniques can be used to identify trends and provide insight on the UN's work reflected in UNGA Resolutions.

In addition to the previously mentioned conventional NLP techniques, recent advancements in deep learning models and their application to NLP methods have yielded highly capable models, one of which is BERT [10]. Upon its introduction, BERT has become a benchmark model for various NLP tasks including text classification, and research efforts on further using BERT for text classification show promising results. Lee and Hsiang [11] proposed PatentBERT, which is a BERT-based model fine-tuned on patent data for patent classification, a multi-label text classification task. PatentBERT was able to achieve a new state-of-the-art result with an $F1$ score of 65.87%. El-Alami et al. [12] used BERT models, including BERT, mBERT, and AraBERT, for a multilingual offensive language detection task. Their findings provided evidence of BERT's robustness in the multilingual text classification with an $F1$ score over 93%. Khan et al. [13] conducted a benchmark study of machine learning models for detecting fake news online. In their study, they analyzed the overall performance of 19 different machine learning models on 3 different datasets, and their findings showed that BERT-based models achieve superior performance on all 3 datasets.

Even though BERT is a highly capable deep learning model, there has been further research on improving its performance, and RoBERTa [14] is one of them. With a few alterations in the architecture and the training procedure of BERT, RoBERTa was able achieve new state-of-the-art results, and the research work on using RoBERTa for text classification proves the model's advancement. Casola et al. [15] conducted an empirical comparison of pre-trained language Transformer models including BERT, RoBERTa, DistillBERT, XLNet, and ALBERT. Their findings showed that on average RoBERTa performed better on text classification. Rodrawangpai and Daungjaiboon [16] were able to surpass the performance of BERT-based models on classifying incident reports by proposing a model architecture based on RoBERTa. Briskilal and Subalalitha [17] combined the capabilities of both BERT and RoBERTa by proposing an ensemble model for the classification of idiomatic and literal sentences. When compared to the base models BERT and RoBERTa alone, their ensemble model achieved a 2% increase in F-score and accuracy.

After the introduction of the 17 SDGs by the UN, many research efforts were also dedicated to identifying and linking SDGs by automatically processing data from various sources. Yeh et al. [18] proposed "SUSTAINBENCH", a benchmark and a public leaderboard website for multiple SDG related datasets with standard train–test splits and well-defined performance metrics. They also provided baseline models and their evaluation results for each dataset. Matsui et al. [19] proposed an NLP model for supporting sustainable development goals, which involves translating semantics, visualizing nexus, and connecting stakeholders. Nilsson et al. [20] developed a framework for mapping interactions between the SDGs. They focused on modeling interactions through important factors, such as geographical context, resource endowments, time horizon, and governance. Smith et al. [21] proposed an approach to quantify the network of SDG interdependencies using policy and scientific documents. Their proposed method combined NLP methods and network analysis to provide a mapping of SDGs' relationships. Toetzke et al. [22] proposed a machine learning framework for categorization of global development aid activities based on textual descriptions. Their framework utilized document embeddings and clustering and generated activity clusters representing the topics of underlying aid activities, many of which were yet to be analyzed empirically. While these works are all based on processing SDG-related data, they focused on different problems than automatically detecting the relevancy of texts in large document collections to one or more specific SDGs.

A limited number of recent works have focused on utilization of machine learning and deep learning techniques to develop text classification systems capable of identifying with high accuracy the related SDGs in a document collection. Pukelis et al. [23] proposed the Open Source SDG (OSDG) project and tool to integrate data from multiple sources into a single framework for SDG classification. The integration aimed to link features from previous approaches and research (e.g., ontology items, keywords, features from machine learning models) to the topics in the Microsoft Academic Graph. Amel-Zadeh et al. [24] provided a proof of concept for the use of ML and NLP to detect companies' alignment with SDGs based on their CSR reports. Their proposed method with binary outcomes used Word2Vec [25] and Doc2Vec models for training a logistic regression classifier, a fully-connected neural network, and an SVM which, with a Doc2Vec [26] embedding, achieved the highest average accuracy of 83.5% for predicting alignment. Guisiano et al. [27,28] proposed a multi-label classification system using BERT and an online tool "SDG-meter" to automate this task. Their proposed BERT model achieved an accuracy of 98%. Despite the high accuracy achieved, the system was only tested on a collection of 400 texts, which is quite limited. Hajikhani and Suominen [29] proposed an ML model to automate the detection of SDG relevancy in patent documents. The authors also presented relatedness between different SDG categories using their highest performing model, which was the logistic regression classifier utilizing Word2Vec. The ML models they utilized achieved above 60% accuracy for most SDGs. While some of the mentioned works have achieved successful results, to the best of our knowledge, none of the existing works evaluated the performance of their models on both binary and multi-class classification tasks for SDG relevance. In addition, fine-tuned RoBERTa, which we demonstrate to outperform all other models in this work, has not been utilized in any of the existing frameworks.

3. Automated Report Processing Framework

In this section, we provide details on our automated report processing framework. We have developed both classical machine learning (ML) and deep learning (DL)-based models for processing of reports to identify relevancy of text blocks to SDGs. Furthermore, we have developed both binary and multi-class classification models. Binary classification models were built to identify whether a text block is relevant to each SDG, whereas multi-class classification models indicate the most relevant SDG for the given text block. The binary classification feature of the framework is an important aid in detecting the presence of different SDGs in documents that may contain text on a variety of topics. The multi-class classification feature is more useful in cases where documents are known to be dedicated to SDGs and when it is necessary to identify which SDG each section is relevant to.

ML and DL-based models involve different processing steps during model building and execution. In the subsections below, we describe the overall operation of each.

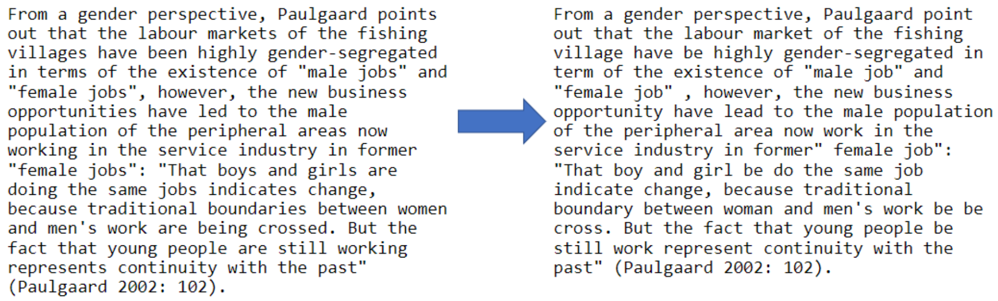
3.1. ML-Based Processing

The automated text processing pipeline for the ML-based methods consists of four stages: text pre-processing, vectorization, model training, and model execution. We start with a training dataset, where each text block is labelled by human annotators as relevant to one of the SDGs.

3.1.1. Pre-Processing

In the text pre-processing stage, we first filter out the records with a low agreement score from the dataset. Having a low agreement score means that multiple annotators had different views about the particular text block being relevant to a specific SDG. Removal of such instances from the data allows us to achieve higher quality in model training, as we will only be learning from instances that we are more certain about. Then, we filter out stop words such as *and*, *the*, *is*, etc., as well as punctuation marks, one-letter words, and numbers, as these do not contribute to the meaning of sentences. To eliminate these, we utilize regular expressions and the relevant methods of the Natural Language Toolkit (NLTK) [30] for Python.

Lemmatization is an operation that converts words into their simplest form. It is widely used in pre-processing of raw text data in NLP tasks. By lemmatizing words, we aim to reduce potential confusions that different representations of the same base word could create, although they add similar or the same meaning to the context. We use NLTK WordNet Lemmatizer for lemmatization of the text. WordNet [31] is a large lexical database consisting of English words, allowing the analysis and processing of English sentences. We first tokenize sentences and find the POS (part-of-speech) tag for each token, which is basically the function of the word in a sentence (noun, verb, adjective, etc.). Then, according to their POS, we find the correct base form of the words using NLTK WordNet Lemmatizer. An example lemmatization is shown in Figure 1.



From a gender perspective, Paulgaard points out that the labour markets of the fishing villages have been highly gender-segregated in terms of the existence of "male jobs" and "female jobs", however, the new business opportunities have led to the male population of the peripheral areas now working in the service industry in former "female jobs": "That boys and girls are doing the same jobs indicates change, because traditional boundaries between women and men's work are being crossed. But the fact that young people are still working represents continuity with the past" (Paulgaard 2002: 102).

From a gender perspective, Paulgaard point out that the labour market of the fishing village have be highly gender-segregated in term of the existence of "male job" and "female job" , however, the new business opportunity have lead to the male population of the peripheral area now work in the service industry in former" female job": "That boy and girl be do the same job indicate change, because traditional boundary between woman and men's work be be cross. But the fact that young people be still work represent continuity with the past" (Paulgaard 2002: 102).

Figure 1. Lemmatization of text.

3.1.2. Vectorization

In order to convert the data into a form that can be processed by ML algorithms, numeric matrices need to be constructed for the data instances. This is achieved through vectorization, which maps each word in the complete dataset to a unique number and keeps counts of the occurrences of each word in the data instances (i.e., text blocks). The importance of a specific word for a particular class should not only be determined by how frequently that word occurs in the data instances of that class, but also by how infrequently it occurs in instances of other classes, i.e., if a word occurs too often in the whole dataset, it carries less information. In order to account for this rationale, we use a TF-IDF vectorizer [32], which calculates a score for each word in a text block by multiplying the frequency score of that word in the text block by the inverse of the frequency of that word for the whole dataset.

3.1.3. Model Training

In this stage, we train different ML models with the dataset formed in the previous stage and optimize the models. In this work, we have utilized four well-known classification algorithms, namely Linear Support Vector Machines, Decision Tree, Logistic Regression, and Gaussian Naive Bayes.

3.2. Deep-Learning-Based Processing

Deep learning (DL) algorithms have achieved significant success in many learning tasks in the past decade with the developments in computing infrastructures and the availability of big data, and the field of NLP is no exception. For the DL-based models, we did not apply any pre-processing to the data because the whole information in a sentence can be helpful when using pre-trained language representation models such as BERT and RoBERTa. They both need special encoding and tokenization methods of the inputs so that they can be trained. For that purpose, we used the dedicated methods in the *Transformers* library of the HuggingFace [33] AI community. During BERT and Roberta experiments, we used BertTokenizer and RobertaTokenizer from HuggingFace's Transformers library to tokenize our text data. Moreover, we used the model configurations called BertForSequenceClassification and RobertaForSequenceClassification in HuggingFace's Transformers library using pre-trained models called bert-large-uncased and roberta-large. We fine-tuned those models with five epochs, using AdamW optimizer and setting the learning rate 1×10^{-5} . We set other parameters as default (betas = (0.9, 0.999), epsilon is 1×10^{-8}, weight decay is 1×10^{-2}). Below, we provide an overview of both DL models utilized.

3.2.1. BERT

BERT [10], which stands for **B**idirectional **E**ncoder **R**epresentations from **T**ransformers, is an open-source, pre-trained deep learning algorithm developed by Google, and it has become a baseline in NLP research. It is proficient at a variety of NLP tasks, including sequence-to-sequence-based language generation tasks, such as question answering and sentence prediction, and natural language understanding tasks, such as sentiment classification and word sense disambiguation.

BERT is designed as an unsupervised model, and it is trained with a large text corpus from the web in a variety of languages and using the masked language modeling (MLM) task which aims to construct outputs from unarranged or corrupted input. BERT's high capability also comes from its bidirectional attention mechanism called *Transformer*, and BERT's model architecture uses Vaswani et al.'s [34] multi-layer bidirectional Transformer encoder. The bidirectional Transformer aims to recognize contextual relationships between words (or subwords) in a text by using the surrounding text to establish context. A Transformer's basic design consists of two independent mechanisms: an encoder that reads the text input and a decoder that generates a word prediction. The encoder consists of a stack of six layers, each of which consists of a multi-head self-attention mechanism and a position-wise fully connected feed-forward neural network. Residual connections are applied around these two layers, followed by a normalization step. The decoder also has six identical layers, which contain a third sub-layer for multi-head attention over the encoder stack's output in addition to the aforementioned two sub-layers [34]. The attention mechanism maps a query and a set of key-value pairs to an output, where the output is calculated as a weighted sum of the values. The weights for each value are computed based on the compatibility of the query with the corresponding key. The details of the attention mechanisms are available at [34].

BERT tokenizes text, then applies a sentence embedding to each token to represent which sentence each word belongs to and a positional embedding to signify the position of the word inside the sentences. Then, the created input embeddings go through the previously mentioned Transformer stacks, where the attention mechanism is applied, normalized, and fed forward.

During training, language models face the challenge of defining a prediction goal. While many language models predict the next word in a sentence, which limits context-based learning, BERT uses two mechanisms, namely Masked Language Modeling (MLM) and next sentence prediction, to better integrate context into the learning process.

The MLM process operates as follows: Before word sequences are fed into BERT, 15% of the words in each sequence are replaced with a [MASK] token whose original values are then predicted by the model based on the context that the non-masked words provide. To achieve this, a classification layer is added on top of the encoder input, the output vectors are multiplied by the embedding matrix to transform them into the vocabulary dimension, and the probability of each word in the vocabulary is calculated using softmax. The loss function of BERT takes into account only the prediction of masked values.

In the next sentence prediction task, the model aims to learn whether in a given pair of sentences, the second sentence is the subsequent sentence to the first. To achieve this, we insert a [CLS] token at the beginning of the initial sentence and a [SEP] token at the end of each sentence. We add a sentence embedding to each token and a positional embedding to each token, which indicates the position of the token in the sequence (as discussed by Vaswani et al. [34]). For the prediction, the whole input sequence is fed into the Transformer model, the output of the [CLS] token is transformed into a vector of size 2×1 with a simple classifier, and the probability of being the next sentence in the sequence is calculated using softmax.

Originally, two BERT models [10] were introduced: BERT-large and BERT-base, both of which are trained on a large text corpus in English gathered from the web. The difference between the two models is number of encoder layers where, in the BERT-base model, there are 12 encoder layers, whereas in the BERT-large, there are 24 encoder layers. In this work, we use BERT-large.

The architecture of BERT-large, as seen in Figure 2, consists of several Transformer encoder stacks, which have a bidirectional nature. This means that BERT learns information from a sequence of words from both directions.

BERT is used for two main approaches: feature extraction and fine-tuning. In feature extraction, the model architecture, including the model parameters, are preserved and used to extract features which can be input for further classifier models. In fine-tuning, on the other hand, the model's architecture is modified by adding one extra layer after the final layer of the original BERT architecture and further training it for more downstream tasks for just a few epochs with additional data specifically prepared for the task. Our work in this paper involves fine-tuning BERT-large with data specific to SDGs.

3.2.2. RoBERTa

Even though BERT achieved advanced performance across various NLP tasks, a number of approaches have further improved BERT's capabilities. One such work is RoBERTa [14], which stands for **R**obustly **O**ptimized **BERT** Pre-training **A**pproach. It is a variation of BERT, proposed by researchers at Facebook and Washington University. RoBERTa not only optimizes the training of BERT architecture during pre-training but also is superior at predicting intentionally concealed sections of a text. RoBERTa uses the same architecture as BERT with a few alterations in the training procedure. RoBERTa alters key hyperparameters in BERT. It removes BERT's next-sentence pre-training intent and just uses MLM instead. In addition to dynamically modifying the masking pattern as opposed to a single static mask in BERT, using a larger batch size (8000), a larger vocabulary (around 50,000 words), higher learning rates, and longer sequences (512 tokens) during training allows RoBERTa to be more successful and more performance-oriented than BERT at masked language modeling.

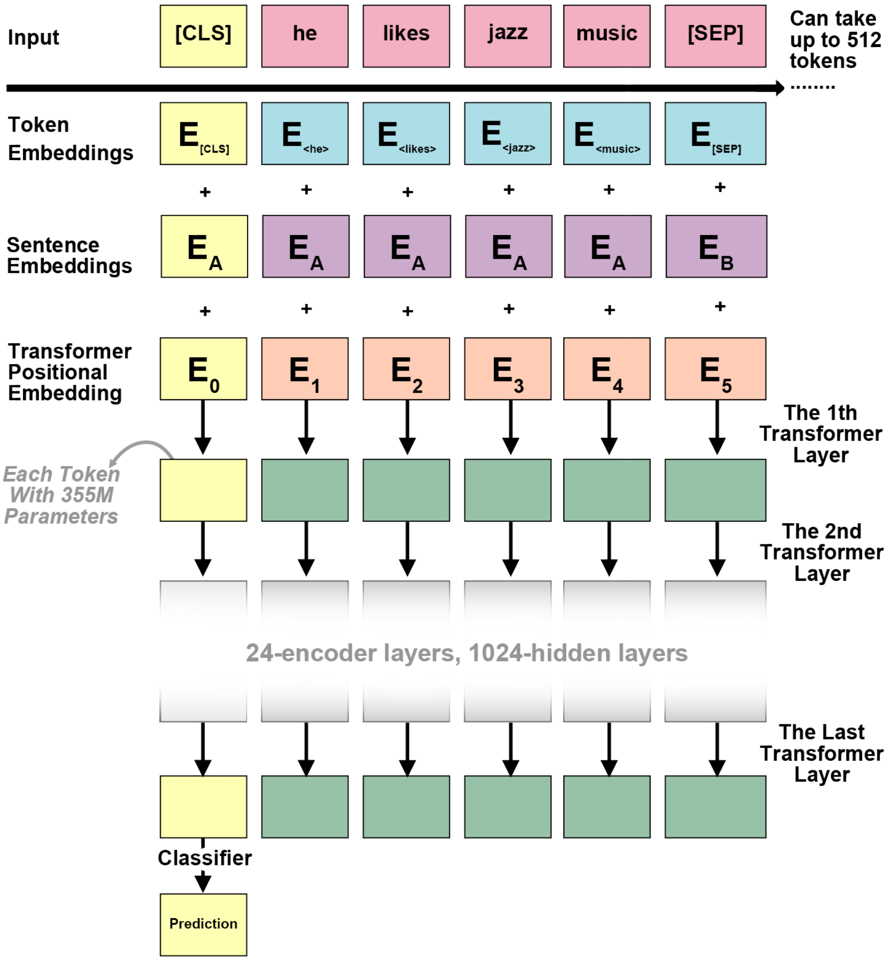


Figure 2. BERT-large Architecture.

4. Evaluation

4.1. SDG Dataset

OSDG Community Dataset (OSDG-CD) [35], first published by the OSDG team on 1 October 2021, is a public dataset that aims to support NLP research and studies on deriving insights into the nature of SDGs. OSDG-CD contains texts with their related SDG labels validated by more than 1000 volunteers from over 100 countries via the OSDG Community Platform (OSDG-CP). In our experiments, we used version 04.2022, which was the latest version of the dataset when this research was conducted. In this section, we provide details of the dataset and present our analysis.

The dataset contains texts from various public documents, such as reports, policy documents, and publication abstracts; moreover, each text excerpt is nearly a paragraph in length. We validated this information by analyzing the word numbers of texts. The longest text consists of 226 tokens (words), while the shortest consists of 16 tokens. Furthermore, the average size of documents is approximately 89.79 tokens per text. Considering these, the dataset can be safely used without truncating to fine-tune most of the Transformer models as they generally consume 512 tokens at maximum.

Although there are 17 Sustainable Development Goals defined by UNDP, the dataset includes the first 15 SDGs. Our observations on the distribution of texts over those SDGs show that the dataset is quite unbalanced. Figure 3 shows the distribution of the SDGs. Therefore, we split the data in a stratified manner so that the ratio between test and train data for each SDG category was preserved.

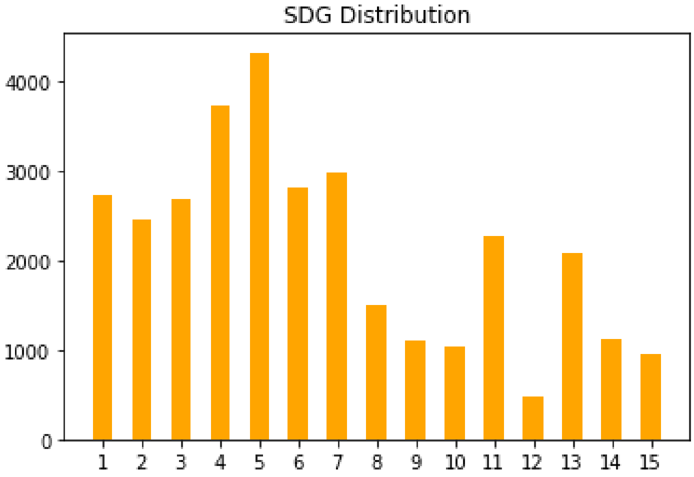


Figure 3. SDG Distribution Histogram.

The volunteers from OSDG-CP contributed to the annotation of the dataset by completing some labeling exercises. Each text was validated by at least three different volunteers and up to nine different volunteers. All the labeling exercises were binary decision problems, meaning that each volunteer could accept or reject a suggested label. This information was embedded into the dataset as *'labels_negative'*, *'labels_positive'*, and *'agreement'* columns, where *'agreement'* represented the agreement score based on the formula below:

$$agreement = \frac{|labels_{positive} - labels_{negative}|}{labels_{positive} + labels_{negative}}$$

When we analyzed the dataset, we observed that some records had low agreement scores. In addition, in some records, the number of volunteers who rejected the suggested label was more than those who accepted it. These 'low-quality' records can potentially reduce the accuracy of classifier models; therefore, it would be appropriate to filter them out in the pre-processing stages of the experiments. Figure 4 shows the distribution of quality and poor quality records in each SDG, where quality records are defined as the records having an agreement score greater than or equal to 0.6 and having *'labels_positive'* greater than *'labels_negative'*. In contrast, the poor quality records are the remaining ones.

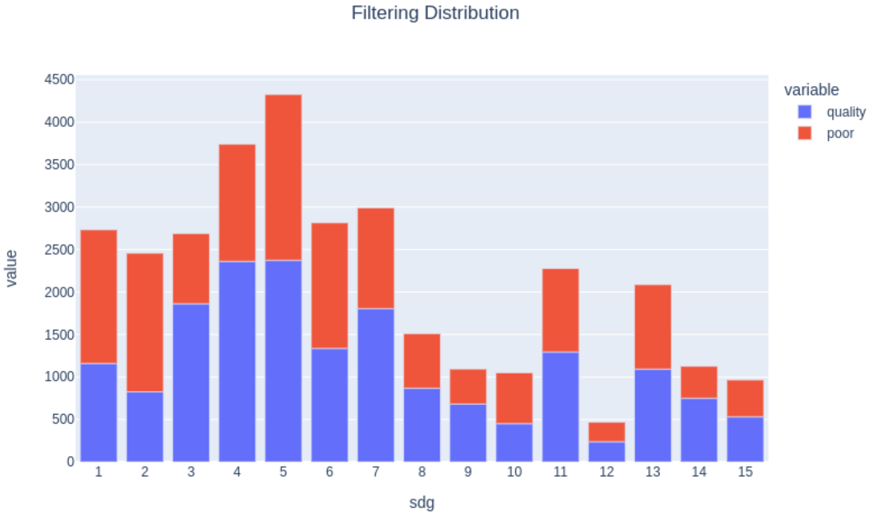


Figure 4. Quality and poor records.

4.2. Evaluation Results

In this section, we report the results of both binary and multi-class classification for the ML and DL-based models. For evaluating the performances of the different algorithms, we utilized commonly used metrics from the ML literature: accuracy, precision, recall, and $F1$ score. These metrics are calculated using the formulae below, where TP is the number of positive instances classified as positive, TN is the number of negative instances classified as negative, FP is the number of negative instances classified as positive, and FN is the number of positive instances classified as negative by the algorithm:

$$Accuracy = \frac{TN + TP}{TN + FP + TP + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4}$$

In the subsections below, we present the performance results of the different models with these metrics. We provide the binary classification ROC (receiver operating characteristic) curves, which plot TP rate vs. FP rate at different decision thresholds, for SVM, BERT, and RoBERTa in Appendix A.

4.2.1. Binary Classification

Foremost, we performed binary classification for all SDGs separately, and while doing this, we created a balanced dataset by undersampling for each SDG group in each trial by taking equal-sized random samples from the ones that do not belong to that SDG. Our goal was to observe what the results would look like if we had a balanced dataset. The number of training and test samples for each SDG in the experiments is shown in Table 1.

Table 1. Train–test split of data in binary classification experiments.

SDG#	Train	Test
1	1856	464
2	1320	330
3	2980	745
4	3780	945
5	3800	950
6	2140	535
7	2892	723
8	1392	348
9	1092	273
10	724	181
11	2072	518
12	384	96
13	1752	438
14	1200	300
15	852	213

Finally, we created a matrix representing each sentence and the words contained in it using the vectorizer we just obtained. Each row of the matrix is a sentence, and each cell is a word that is tokenized. At the model training stage for ML algorithms, we trained four models for all SDGs:

- Linear Support Vector Machines (SVM);
- Decision Tree;
- Logistic Regression (LR);
- Gaussian Naive Bayes.

We used the parameters of the algorithms provided in Python's 'scikit-learn' library, as listed in Table 2.

Table 2. Parameter values for ML models.

SVM:		
C: 1.0	kernel: 'linear'	degree: 3
gamma: 'scale'	coef0: 0.0	shrinking: True
probability: True	tol = 1×10^{-3}	cache_size: 200
class_weight: None	verbose: False	max_iter: -1
decision_function_shape: 'ovr'	break_ties: False	random_state: 42
Decision Tree Classifier:		
criterion: "gini"	splitter: "best"	max_depth: None
min_samples_split: 2	min_samples_leaf: 1	min_weight_fraction_leaf: 0.0
max_features: None	random_state: None	max_leaf_nodes: None
min_impurity_decrease: 0.0	class_weight: None	ccp_alpha 0.0
Logistic Regression:		
penalty: 'l2'	dual: False	tol: 1×10^{-4}
C: 1.0	fit_intercept: True	intercept_scaling: 1
class_weight: None	random_state: 0	solver: 'lbfgs'
max_iter: 100	multi_class: 'auto'	verbose: 0
warm_start: False	n_jobs: None	l1_ratio: None
Gaussian Naive Bayes:		
priors: - (Not used)	var_smoothing: 1×10^{-9}	

While constructing the SVM model, we chose the C-Support Vector Classification variant (SVC). We applied five-fold cross-validation while evaluating all models, which is a common practice for validating the performance of ML and DL models. This validation process can be explained simply as follows: The dataset is randomly split into five 'folds', and each fold is used as the test dataset for the validation of the models in different iterations, while the models are trained with the remaining four folds in each iteration. At the end of five iterations, the averages of the evaluation results of those five iterations are calculated. Table 3 summarizes the $F1$ score results of the four ML algorithms for each SDG.

At the model evaluation stage for ML models, we concluded that SVM and Logistic Regression models produce the best results. Although these models achieved relatively better results than the other traditional ML algorithms, we applied DL-based NLP methods for further comparison. First, we trained BERT and RoBERTa models with the same train and test datasets and then evaluated their binary classification performance. The hyperparameters we set for both models can be seen in Table 4 below.

Table 3. Binary classification results ($F1$ scores) for machine learning models.

SDG#	LR	SVM	Naive Bayes	Decision Tree
1	0.93	0.94	0.78	0.79
2	0.94	0.94	0.79	0.90
3	0.96	0.97	0.85	0.93
4	0.97	0.97	0.82	0.95
5	0.97	0.98	0.79	0.96
6	0.96	0.96	0.79	0.95
7	0.95	0.96	0.81	0.93
8	0.89	0.89	0.73	0.78
9	0.93	0.92	0.77	0.82
10	0.86	0.88	0.78	0.77
11	0.94	0.94	0.78	0.87
12	0.90	0.90	0.78	0.86
13	0.94	0.96	0.81	0.92
14	0.97	0.97	0.86	0.96
15	0.94	0.95	0.83	0.88

Table 4. Parameter values for DL models.

BERT:	
Model: BertForSequenceClassification batch_size: 3 Learning rate (lr): 1×10^{-5} Weights: 'bert-large-uncased' Tokenizer: BertTokenizer	num_labels: 15 optimizer: AdamW Epsilon(eps): 1×10^{-8} epochs: 5
RoBERTa:	
Model: RobertaForSequenceClassification batch_size: 3 Learning rate (lr): 1×10^{-5} Weights: 'roberta-large' Tokenizer: RobertaTokenizer	num_labels: 15 optimizer: AdamW Epsilon(eps): 1×10^{-8} epochs: 5

Note: num_labels is set to 2 during binary classification experiments.

As expected, these models outperformed the traditional ML algorithms. The results (rounded to the nearest hundredth) can be seen in Tables 5 and 6 for BERT and RoBERTa, respectively. While DL models achieved significantly better performance over ML models in this task, this comes with a cost in training time and model size. The sizes of fine-tuned BERT binary models are around 1.25 GB, and the sizes of fine-tuned RoBERTa binary models are around 1.32 GB, while the sizes of all ML models are in the order of KB. The fine-tuning of BERT and RoBERTa for each classifier took approximately 4 h in the Google collaborative environment [36] using available GPUs, while the time to train ML models was in the order of seconds. Since model training is only performed once, the training time cost is tolerable when the significant accuracy increase is considered.

Table 5. Binary classification results for BERT.

SDG#	Accuracy	Precision	Recall	*F*1
1	0.95	0.95	0.95	0.95
2	0.97	0.98	0.97	0.97
3	0.99	0.98	0.99	0.99
4	0.98	0.99	0.98	0.98
5	0.99	0.99	0.99	0.99
6	0.98	0.99	0.97	0.98
7	0.98	0.98	0.98	0.98
8	0.90	0.91	0.89	0.90
9	0.95	0.96	0.94	0.95
10	0.89	0.88	0.90	0.89
11	0.96	0.96	0.97	0.96
12	0.94	0.96	0.92	0.94
13	0.96	0.96	0.96	0.96
14	0.99	0.99	0.99	0.99
15	0.97	0.97	0.96	0.97

Table 6. Binary classification results for RoBERTa.

SDG#	Accuracy	Precision	Recall	*F*1
1	0.96	0.97	0.94	0.96
2	0.97	0.98	0.96	0.97
3	0.99	0.98	0.99	0.99
4	0.98	0.99	0.98	0.98
5	0.99	0.99	0.98	0.99
6	0.98	0.99	0.98	0.98
7	0.98	0.99	0.97	0.98
8	0.92	0.91	0.92	0.92
9	0.95	0.96	0.95	0.95
10	0.89	0.90	0.88	0.89
11	0.96	0.97	0.96	0.96
12	0.96	0.98	0.94	0.96
13	0.97	0.97	0.96	0.97
14	0.99	0.99	0.99	0.99
15	0.97	0.96	0.97	0.97

4.2.2. Multi-Class Classification

In this section, we present the results of our multi-class classification experiments. As before, these experiments consist of two sub-experiments. The first uses supervised ML methods, and the second fine-tunes BERT and RoBERTa models. As before, five-fold cross validation was performed in the experiments. Table 7 shows the number of training and test instances for each SDG in the experiments.

Table 7. Train–test split of data in multi-class classification experiments.

SDG#	Train	Test
1	928	232
2	660	165
3	1492	373
4	1892	473
5	1900	475
6	1068	267
7	1444	361
8	692	173
9	548	137
10	360	90
11	1036	259
12	188	47
13	872	218
14	600	150
15	428	107

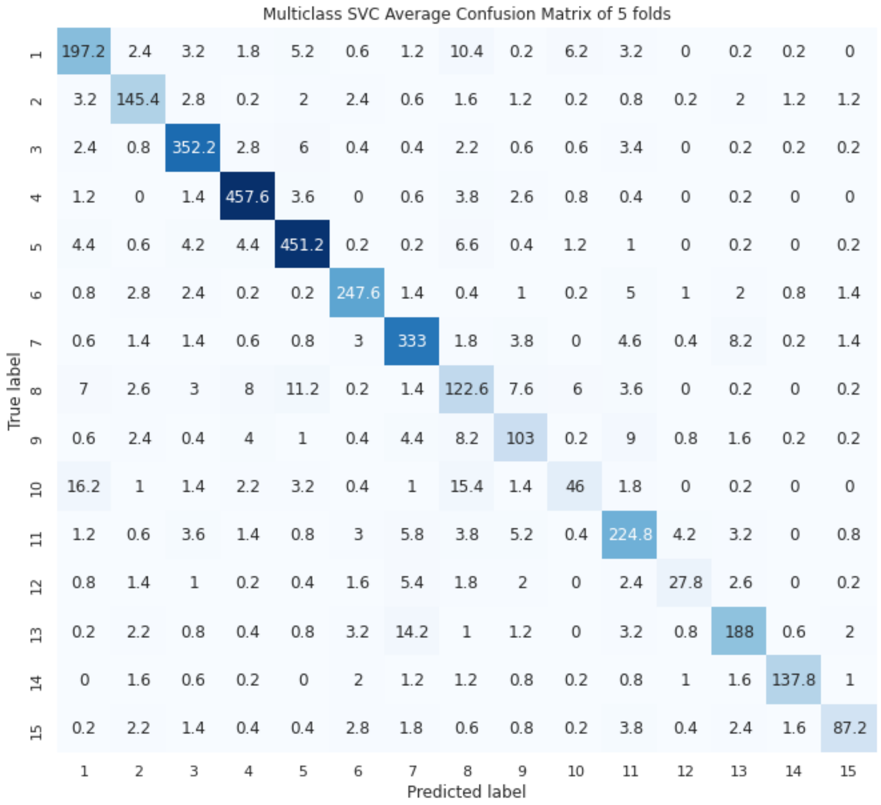
For multi-class classification models, in addition to $F1$ score, we also used confusion matrices to demonstrate the classification algorithms' performance. Since we have more than two classes in these classification tasks, by plotting a confusion matrix, we can better see which SDGs our models are confusing the other SDGs with. A confusion matrix is used to demonstrate how many test samples of a specific class were classified as instances of all classes in our task. For example, in the SVC confusion matrix in Figure 5, by looking at the first row, we see that for SDG1, the classifier correctly predicted 197 instances as belonging to class SDG1, while it incorrectly predicted two samples as belonging to class SDG2.

The $F1$ scores and confusion matrices based on the performances of the mentioned ML models can be seen in Figures 5–8. As seen in the confusion matrices, the lack of sufficient data for SDG10: Reduced Inequalities and SDG12: Responsible Consumption and Production, have negatively affected the models' performance.

The pseudocode of the training and evaluation steps of multiclass classification using BERT and RoBERTa are provided in Algorithms 1 and 2.

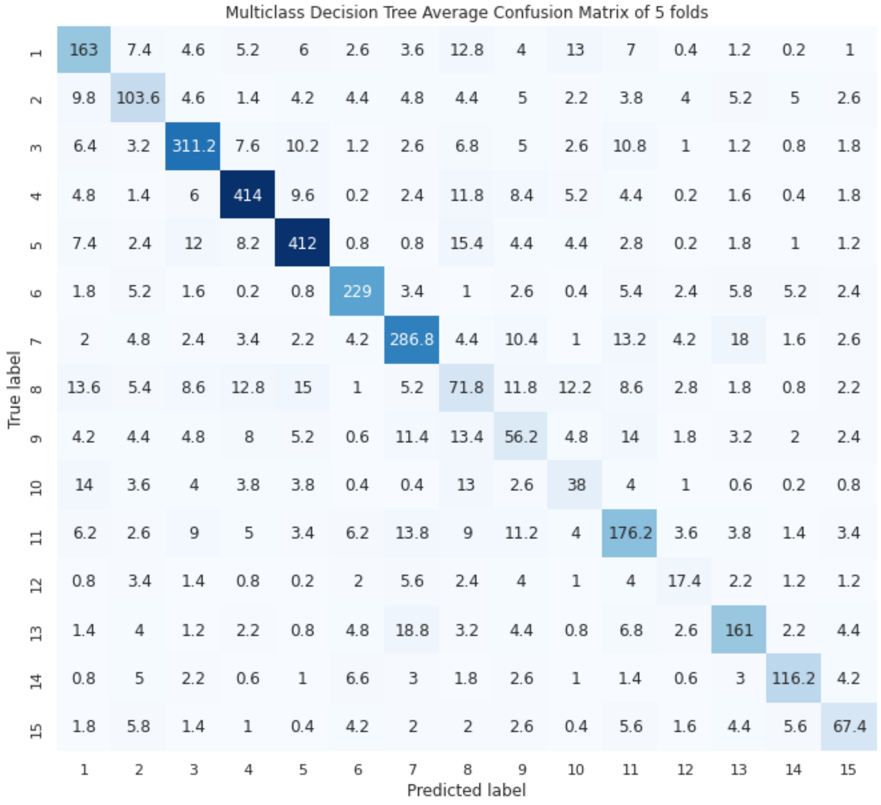
Encoding using a tokenizer (tokenizer.encode_batch_plus) includes the following steps:

1. Add special tokens to tokenized texts:
   - $[CLS]$ at the beginning of each sentence;
   - $[SEP]$ at the end of each sentence.
2. Make each sentence the same length (512 tokens):
   - Add padding ($[PAD]$) tokens to shorter sentences;
   - Truncate longer sentences from the end.
3. Create attention masks:
   - Create a list for each tokenized sentence consisting of zeroes for padding tokens and ones for regular tokens to prevent the model from performing attention on padding token.s



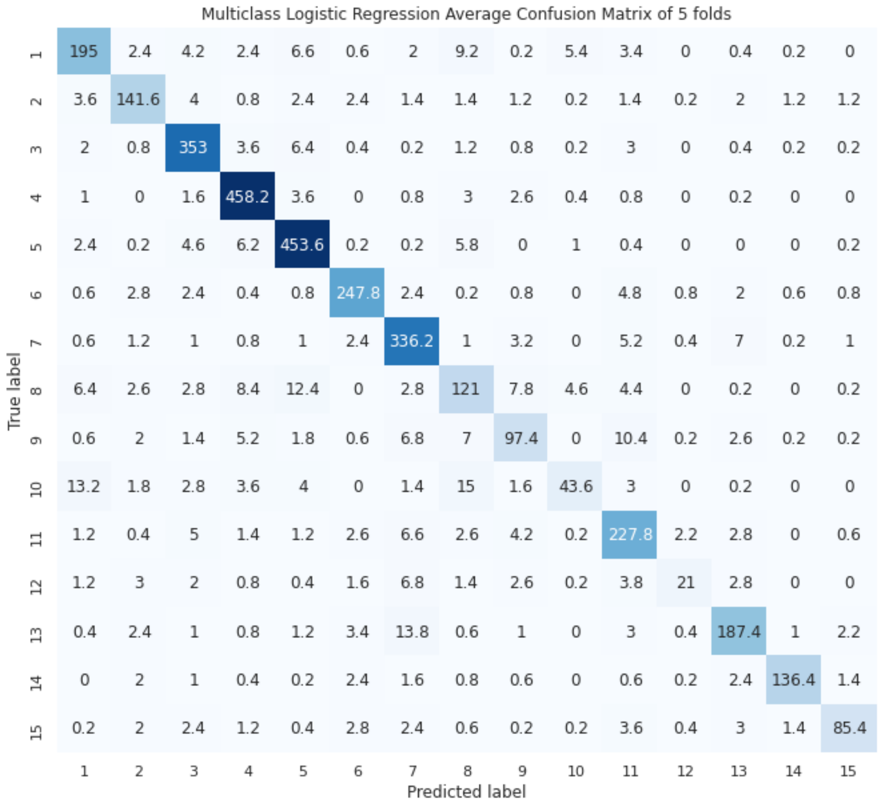
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	197.2	2.4	3.2	1.8	5.2	0.6	1.2	10.4	0.2	6.2	3.2	0	0.2	0.2	0
2	3.2	145.4	2.8	0.2	2	2.4	0.6	1.6	1.2	0.2	0.8	0.2	2	1.2	1.2
3	2.4	0.8	352.2	2.8	6	0.4	0.4	2.2	0.6	0.6	3.4	0	0.2	0.2	0.2
4	1.2	0	1.4	457.6	3.6	0	0.6	3.8	2.6	0.8	0.4	0	0.2	0	0
5	4.4	0.6	4.2	4.4	451.2	0.2	0.2	6.6	0.4	1.2	1	0	0.2	0	0.2
6	0.8	2.8	2.4	0.2	0.2	247.6	1.4	0.4	1	0.2	5	1	2	0.8	1.4
7	0.6	1.4	1.4	0.6	0.8	3	333	1.8	3.8	0	4.6	0.4	8.2	0.2	1.4
8	7	2.6	3	8	11.2	0.2	1.4	122.6	7.6	6	3.6	0	0.2	0	0.2
9	0.6	2.4	0.4	4	1	0.4	4.4	8.2	103	0.2	9	0.8	1.6	0.2	0.2
10	16.2	1	1.4	2.2	3.2	0.4	1	15.4	1.4	46	1.8	0	0.2	0	0
11	1.2	0.6	3.6	1.4	0.8	3	5.8	3.8	5.2	0.4	224.8	4.2	3.2	0	0.8
12	0.8	1.4	1	0.2	0.4	1.6	5.4	1.8	2	0	2.4	27.8	2.6	0	0.2
13	0.2	2.2	0.8	0.4	0.8	3.2	14.2	1	1.2	0	3.2	0.8	188	0.6	2
14	0	1.6	0.6	0.2	0	2	1.2	1.2	0.8	0.2	0.8	1	1.6	137.8	1
15	0.2	2.2	1.4	0.4	0.4	2.8	1.8	0.6	0.8	0.2	3.8	0.4	2.4	1.6	87.2

Figure 5. SVC confusion matrix for the average of 5 folds. *F*1 Score: 0.89.



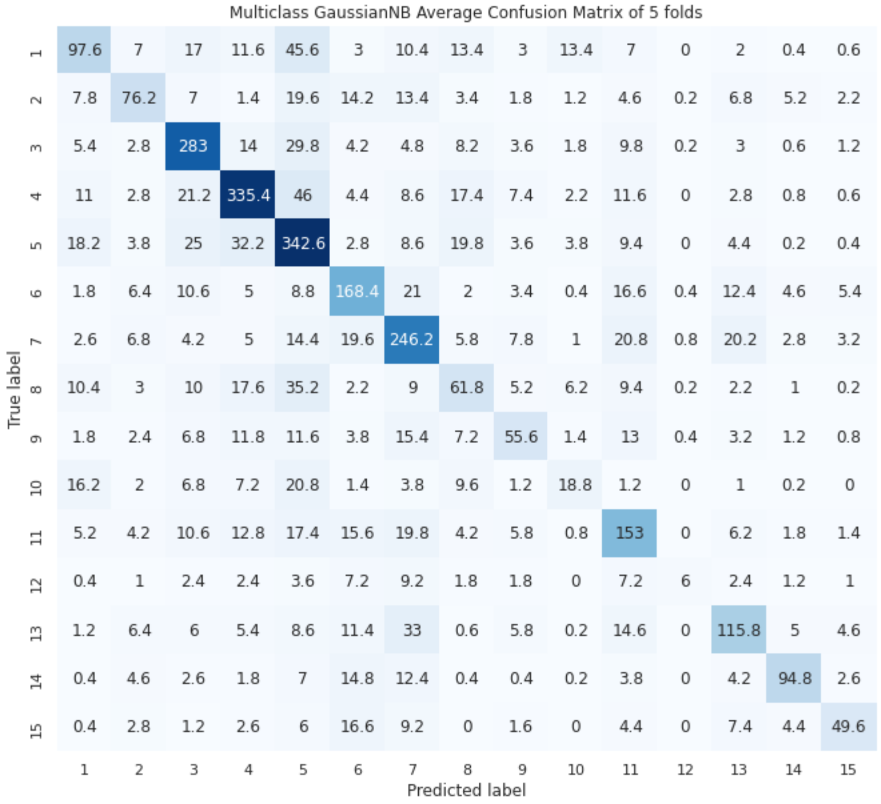
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	163	7.4	4.6	5.2	6	2.6	3.6	12.8	4	13	7	0.4	1.2	0.2	1
2	9.8	103.6	4.6	1.4	4.2	4.4	4.8	4.4	5	2.2	3.8	4	5.2	5	2.6
3	6.4	3.2	311.2	7.6	10.2	1.2	2.6	6.8	5	2.6	10.8	1	1.2	0.8	1.8
4	4.8	1.4	6	414	9.6	0.2	2.4	11.8	8.4	5.2	4.4	0.2	1.6	0.4	1.8
5	7.4	2.4	12	8.2	412	0.8	0.8	15.4	4.4	4.4	2.8	0.2	1.8	1	1.2
6	1.8	5.2	1.6	0.2	0.8	229	3.4	1	2.6	0.4	5.4	2.4	5.8	5.2	2.4
7	2	4.8	2.4	3.4	2.2	4.2	286.8	4.4	10.4	1	13.2	4.2	18	1.6	2.6
8	13.6	5.4	8.6	12.8	15	1	5.2	71.8	11.8	12.2	8.6	2.8	1.8	0.8	2.2
9	4.2	4.4	4.8	8	5.2	0.6	11.4	13.4	56.2	4.8	14	1.8	3.2	2	2.4
10	14	3.6	4	3.8	3.8	0.4	0.4	13	2.6	38	4	1	0.6	0.2	0.8
11	6.2	2.6	9	5	3.4	6.2	13.8	9	11.2	4	176.2	3.6	3.8	1.4	3.4
12	0.8	3.4	1.4	0.8	0.2	2	5.6	2.4	4	1	4	17.4	2.2	1.2	1.2
13	1.4	4	1.2	2.2	0.8	4.8	18.8	3.2	4.4	0.8	6.8	2.6	161	2.2	4.4
14	0.8	5	2.2	0.6	1	6.6	3	1.8	2.6	1	1.4	0.6	3	116.2	4.2
15	1.8	5.8	1.4	1	0.4	4.2	2	2	2.6	0.4	5.6	1.6	4.4	5.6	67.4

Figure 6. Decision tree confusion matrix for the average of 5 folds. *F*1 Score: 0.74.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	195	2.4	4.2	2.4	6.6	0.6	2	9.2	0.2	5.4	3.4	0	0.4	0.2	0
2	3.6	141.6	4	0.8	2.4	2.4	1.4	1.4	1.2	0.2	1.4	0.2	2	1.2	1.2
3	2	0.8	353	3.6	6.4	0.4	0.2	1.2	0.8	0.2	3	0	0.4	0.2	0.2
4	1	0	1.6	458.2	3.6	0	0.8	3	2.6	0.4	0.8	0	0.2	0	0
5	2.4	0.2	4.6	6.2	453.6	0.2	0.2	5.8	0	1	0.4	0	0	0	0.2
6	0.6	2.8	2.4	0.4	0.8	247.8	2.4	0.2	0.8	0	4.8	0.8	2	0.6	0.8
7	0.6	1.2	1	0.8	1	2.4	336.2	1	3.2	0	5.2	0.4	7	0.2	1
8	6.4	2.6	2.8	8.4	12.4	0	2.8	121	7.8	4.6	4.4	0	0.2	0	0.2
9	0.6	2	1.4	5.2	1.8	0.6	6.8	7	97.4	0	10.4	0.2	2.6	0.2	0.2
10	13.2	1.8	2.8	3.6	4	0	1.4	15	1.6	43.6	3	0	0.2	0	0
11	1.2	0.4	5	1.4	1.2	2.6	6.6	2.6	4.2	0.2	227.8	2.2	2.8	0	0.6
12	1.2	3	2	0.8	0.4	1.6	6.8	1.4	2.6	0.2	3.8	21	2.8	0	0
13	0.4	2.4	1	0.8	1.2	3.4	13.8	0.6	1	0	3	0.4	187.4	1	2.2
14	0	2	1	0.4	0.2	2.4	1.6	0.8	0.6	0	0.6	0.2	2.4	136.4	1.4
15	0.2	2	2.4	1.2	0.4	2.8	2.4	0.6	0.2	0.2	3.6	0.4	3	1.4	85.4

Figure 7. Logistic regression confusion matrix for the average of 5 folds. *F*1 Score: 0.88.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	97.6	7	17	11.6	45.6	3	10.4	13.4	3	13.4	7	0	2	0.4	0.6
2	7.8	76.2	7	1.4	19.6	14.2	13.4	3.4	1.8	1.2	4.6	0.2	6.8	5.2	2.2
3	5.4	2.8	283	14	29.8	4.2	4.8	8.2	3.6	1.8	9.8	0.2	3	0.6	1.2
4	11	2.8	21.2	335.4	46	4.4	8.6	17.4	7.4	2.2	11.6	0	2.8	0.8	0.6
5	18.2	3.8	25	32.2	342.6	2.8	8.6	19.8	3.6	3.8	9.4	0	4.4	0.2	0.4
6	1.8	6.4	10.6	5	8.8	168.4	21	2	3.4	0.4	16.6	0.4	12.4	4.6	5.4
7	2.6	6.8	4.2	5	14.4	19.6	246.2	5.8	7.8	1	20.8	0.8	20.2	2.8	3.2
8	10.4	3	10	17.6	35.2	2.2	9	61.8	5.2	6.2	9.4	0.2	2.2	1	0.2
9	1.8	2.4	6.8	11.8	11.6	3.8	15.4	7.2	55.6	1.4	13	0.4	3.2	1.2	0.8
10	16.2	2	6.8	7.2	20.8	1.4	3.8	9.6	1.2	18.8	1.2	0	1	0.2	0
11	5.2	4.2	10.6	12.8	17.4	15.6	19.8	4.2	5.8	0.8	153	0	6.2	1.8	1.4
12	0.4	1	2.4	2.4	3.6	7.2	9.2	1.8	1.8	0	7.2	6	2.4	1.2	1
13	1.2	6.4	6	5.4	8.6	11.4	33	0.6	5.8	0.2	14.6	0	115.8	5	4.6
14	0.4	4.6	2.6	1.8	7	14.8	12.4	0.4	0.4	0.2	3.8	0	4.2	94.8	2.6
15	0.4	2.8	1.2	2.6	6	16.6	9.2	0	1.6	0	4.4	0	7.4	4.4	49.6

Figure 8. Gaussian Naive Bayes confusion matrix for the average of 5 folds. *F*1 Score: 0.60.

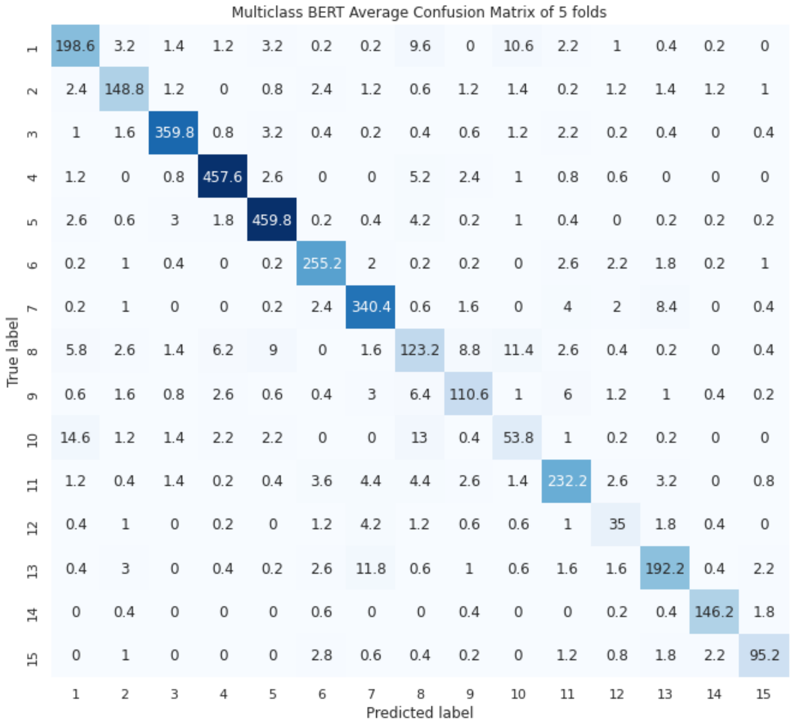
Algorithm 1: Five-fold Multiclass Classification Using fine-tuned BERT

```
tokenizer ← BertTokenizer
folds ← five folds of the dataset, stratified by labels (SDGs)
for fold in folds do
    model ← BertForSequenceClassification(“bert-large-uncased”, num_labels=15)

    $X_{test}$ ← texts in this fold
    $Y_{test}$ ← labels (SDGs) of texts in $X_{test}$
    $X_{train}$ ← texts in other folds
    $Y_{train}$ ← labels (SDGs) of texts in $X_{train}$

    encoded_train_data ← tokenizer.batch_encode_plus($X_{train}$)

    encoded_test_data ← tokenizer.batch_encode_plus($X_{test}$)

    dataset_train ← TensorDataset(encoded_train_data)
    dataloader_train ← DataLoader(dataset_train, batch_size=3)

    dataset_test ← TensorDataset(encoded_test_data)
    dataloader_test ← DataLoader(dataset_test, batch_size=3)

    //Set AdamW optimizer parameters:
    Learning rate (lr) ← $1 \times 10^{-5}$
    Epsilon (eps) ← $1 \times 10^{-8}$

    for epoch in range(5) do
        //Prepare model for training
        for batch in dataloader_train do
            Feed model with input_ids, attention_mask and labels in
              dataloader_train
            Compute loss and gradients from outputs of the model using
              loss.backward()
            Apply backpropagation using optimizer.step()
        end
        //Prepare model for testing
        Feed model with input_ids, attention_masks and labels in dataloader_test
        Fetch logits (the vector of raw predictions) from the outputs of the model
        Compare logits with true labels of labels in dataloader_test and derive
          evaluation scores (Recall, Precision and $F1$ Score)
        Store the evaluation results of this fold and this epoch
    end
end
```

Algorithm 2: Five-fold Multiclass Classification Using fine-tuned RoBERTa

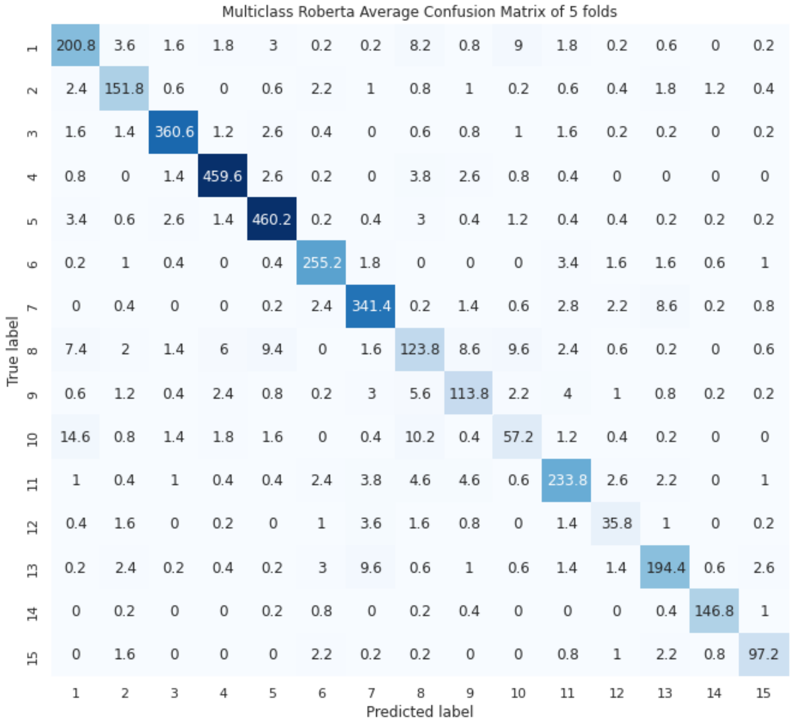
```
tokenizer ← RobertaTokenizer
folds ← five folds of the dataset, stratified by labels (SDGs)
for fold in folds do
    model ← RobertaForSequenceClassification(“roberta-large”, num_labels=15)

    $X_{test}$ ← texts in this fold
    $Y_{test}$ ← labels (SDGs) of texts in $X_{test}$
    $X_{train}$ ← texts in other folds
    $Y_{train}$ ← labels (SDGs) of texts in $X_{train}$

    encoded_train_data ← tokenizer.batch_encode_plus($X_{train}$)

    encoded_test_data ← tokenizer.batch_encode_plus($X_{test}$)

    dataset_train ← TensorDataset(encoded_train_data)
    dataloader_train ← DataLoader(dataset_train, batch_size=3)

    dataset_test ← TensorDataset(encoded_test_data)
    dataloader_test ← DataLoader(dataset_test, batch_size=3)

    //Set AdamW optimizer parameters:
    Learning rate ($lr$) ← $1 \times 10^{-5}$
    Epsilon ($eps$) ← $1 \times 10^{-8}$

    for epoch in range(5) do
        //Prepare model for training
        for batch in dataloader_train do
            Feed model with input_ids, attention_mask and labels in
              dataloader_train
            Compute loss and gradients from outputs of the model using
              loss.backward()
            Apply backpropagation using optimizer.step()
        end
        //Prepare model for testing
        Feed model with input_ids, attention_masks and labels in dataloader_test
        Fetch logits (the vector of raw predictions) from the outputs of the model
        Compare logits with true labels of labels in dataloader_test and derive
          evaluation scores (Recall, Precision and $F1$ Score)
        Store the evaluation results of this fold and this epoch.
    end
end
```

After training the models and evaluating their performances, we found that BERT and RoBERTa outperformed the traditional ML methods that we tried in the first sub-experiment. Their $F1$ scores and confusion matrices can be seen in Figures 9 and 10. We achieved an $F1$ score of 91 percent with BERT and 92 percent with RoBERTa, which are quite high. These results demonstrate that especially deep learning-based NLP models achieve significant success in the attempted task, which is promising for automated processing of large collections of sustainability reports for detection of relevance to SDGs.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	198.6	3.2	1.4	1.2	3.2	0.2	0.2	9.6	0	10.6	2.2	1	0.4	0.2	0
2	2.4	148.8	1.2	0	0.8	2.4	1.2	0.6	1.2	1.4	0.2	1.2	1.4	1.2	1
3	1	1.6	359.8	0.8	3.2	0.4	0.2	0.4	0.6	1.2	2.2	0.2	0.4	0	0.4
4	1.2	0	0.8	457.6	2.6	0	0	5.2	2.4	1	0.8	0.6	0	0	0
5	2.6	0.6	3	1.8	459.8	0.2	0.4	4.2	0.2	1	0.4	0	0.2	0.2	0.2
6	0.2	1	0.4	0	0.2	255.2	2	0.2	0.2	0	2.6	2.2	1.8	0.2	1
7	0.2	1	0	0	0.2	2.4	340.4	0.6	1.6	0	4	2	8.4	0	0.4
8	5.8	2.6	1.4	6.2	9	0	1.6	123.2	8.8	11.4	2.6	0.4	0.2	0	0.4
9	0.6	1.6	0.8	2.6	0.6	0.4	3	6.4	110.6	1	6	1.2	1	0.4	0.2
10	14.6	1.2	1.4	2.2	2.2	0	0	13	0.4	53.8	1	0.2	0.2	0	0
11	1.2	0.4	1.4	0.2	0.4	3.6	4.4	4.4	2.6	1.4	232.2	2.6	3.2	0	0.8
12	0.4	1	0	0.2	0	1.2	4.2	1.2	0.6	0.6	1	35	1.8	0.4	0
13	0.4	3	0	0.4	0.2	2.6	11.8	0.6	1	0.6	1.6	1.6	192.2	0.4	2.2
14	0	0.4	0	0	0	0.6	0	0	0.4	0	0	0.2	0.4	146.2	1.8
15	0	1	0	0	0	2.8	0.6	0.4	0.2	0	1.2	0.8	1.8	2.2	95.2

Figure 9. BERT confusion matrix for average of 5 folds. *F*1 Score: 0.91.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	200.8	3.6	1.6	1.8	3	0.2	0.2	8.2	0.8	9	1.8	0.2	0.6	0	0.2
2	2.4	151.8	0.6	0	0.6	2.2	1	0.8	1	0.2	0.6	0.4	1.8	1.2	0.4
3	1.6	1.4	360.6	1.2	2.6	0.4	0	0.6	0.8	1	1.6	0.2	0.2	0	0.2
4	0.8	0	1.4	459.6	2.6	0.2	0	3.8	2.6	0.8	0.4	0	0	0	0
5	3.4	0.6	2.6	1.4	460.2	0.2	0.4	3	0.4	1.2	0.4	0.4	0.2	0.2	0.2
6	0.2	1	0.4	0	0.4	255.2	1.8	0	0	0	3.4	1.6	1.6	0.6	1
7	0	0.4	0	0	0.2	2.4	341.4	0.2	1.4	0.6	2.8	2.2	8.6	0.2	0.8
8	7.4	2	1.4	6	9.4	0	1.6	123.8	8.6	9.6	2.4	0.6	0.2	0	0.6
9	0.6	1.2	0.4	2.4	0.8	0.2	3	5.6	113.8	2.2	4	1	0.8	0.2	0.2
10	14.6	0.8	1.4	1.8	1.6	0	0.4	10.2	0.4	57.2	1.2	0.4	0.2	0	0
11	1	0.4	1	0.4	0.4	2.4	3.8	4.6	4.6	0.6	233.8	2.6	2.2	0	1
12	0.4	1.6	0	0.2	0	1	3.6	1.6	0.8	0	1.4	35.8	1	0	0.2
13	0.2	2.4	0.2	0.4	0.2	3	9.6	0.6	1	0.6	1.4	1.4	194.4	0.6	2.6
14	0	0.2	0	0	0.2	0.8	0	0.2	0.4	0	0	0	0.4	146.8	1
15	0	1.6	0	0	0	2.2	0.2	0.2	0	0	0.8	1	2.2	0.8	97.2

Figure 10. RoBERTa confusion matrix for average of 5 folds. *F*1 Score: 0.92.

5. Conclusions

It is well established that the SDGs play a key role in the strategic objectives of diverse entities. Nevertheless, connecting projects and activities to the SDGs has been rather complicated and not always possible with existing methods. NLP provides a novel way to classify linkages for SDGs from text data. This research examined various machine learning and deep learning approaches optimized for NLP text classification tasks for their success in classifying textual data according to their relevance to SDGs. Extensive experiments were performed with the recently released OSDG Community Dataset. Results demonstrate that the fine-tuned RoBERTa-based classification models we built, which we have made publicly available, achieve significant success in the attempted task, which is promising for automated processing of large document collections for detection of relevance to SDGs. The framework we have developed in this work can be readily used by the community for processing sustainability reports with high SDG detection/identification accuracy, making it an important contribution to the field. In our future work, we aim to use the same methodology to classify national artificial intelligence (AI) strategy documents of over 50 countries to examine the lineage between SDGs and AI development, particularly in the Global South.

Author Contributions: Conceptualization, M.A. (Merih Angin) and G.D. (Gökhan Dikmener); methodology, M.A. (Merih Angin) and P.A.; software, B.T., C.A.Y., and G.D. (Gökcan Demiralp); validation, B.T., C.A.Y., and G.D. (Gökcan Demiralp); formal analysis, B.T., C.A.Y., and G.D. (Gökcan Demiralp); investigation, M.A. (Merih Angin) and P.A.; resources, M.A. (Merih Angin); data curation, B.T., C.A.Y., G.D. (Gökcan Demiralp), and G.D. (Gökhan Dikmener); writing—original draft preparation, M.A. (Merih Angin), P.A., B.T., C.A.Y., G.D. (Gökcan Demiralp), and M.A. (Mert Atay); writing—review and editing, M.A. (Merih Angin), P.A., M.A. (Mert Atay) and G.D. (Gökhan Dikmener); visualization, B.T., C.A.Y., and G.D. (Gökcan Demiralp); supervision, M.A. (Merih Angin), P.A., and G.D. (Gökhan Dikmener); project administration, M.A. (Merih Angin) and P.A.; funding acquisition, M.A. (Merih Angin). All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by H2020 Marie Skłodowska-Curie Actions (H2020-MSCA-IF-2019) grant number 896716. The funder had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The OSDG Community Dataset (OSDG-CD) used in this study is publicly available via https://zenodo.org/record/6393942#.Yy9bY1JBzqp (accessed on 24 September 2022). The RoBERTa classification models we have developed, as well as the training and evaluation code are available for public use at https://ma-cssl.ku.edu.tr/research/ (accessed on 24 September 2022).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
BERT	Bidirectional Encoder Representations from Transformers
CSR	Corporate social responsibility
DL	Deep Learning
ESG	Environmental, Social and Governance
GRI	Global Reporting Initiative
ML	Machine Learning
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
OSDG	Open Source SDG
OSDG-CP	OSDG Community Platform
POS	Part of speech
RoBERTa	Robustly Optimized BERT Pre-training Approach
SDG	Sustainable Development Goal
TF-IDF	Term Frequency-Inverse Document Frequency
UN	United Nations
UNDP	United Nations Development Program
UNGA	United Nations General Assembly

Appendix A

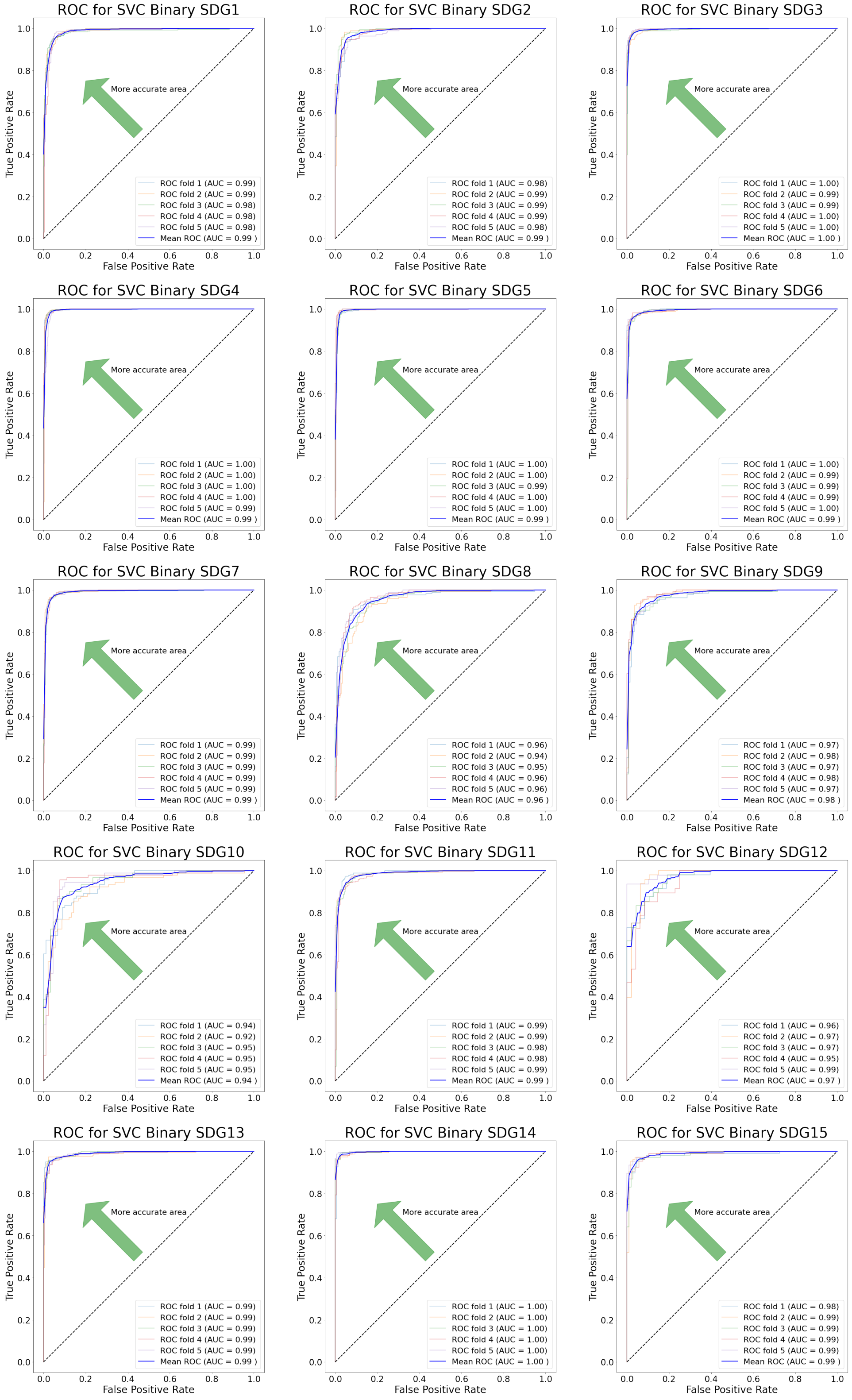


Figure A1. Support vector classification ROC curves for each fold.

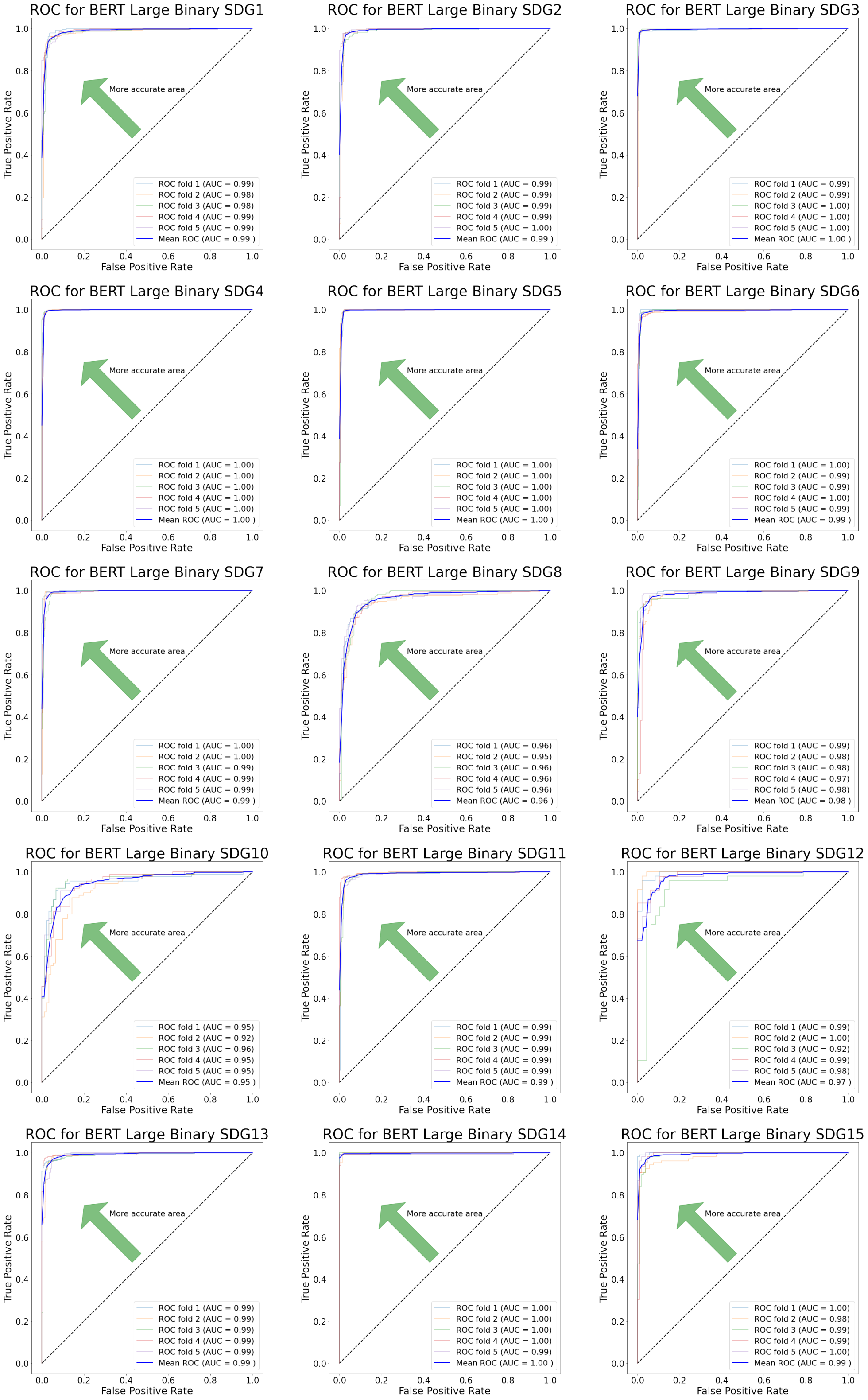


Figure A2. BERT ROC curves for each fold.

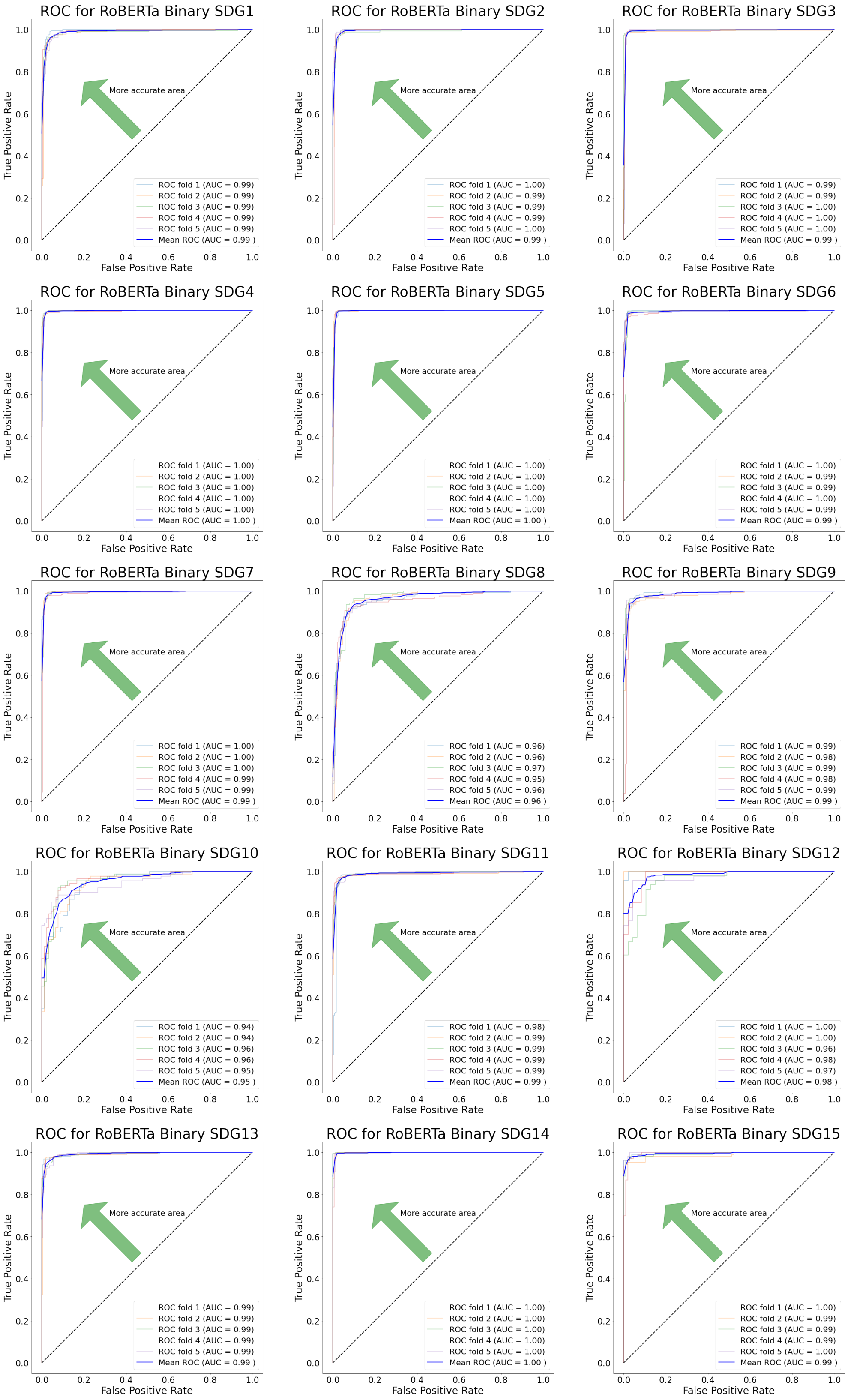


Figure A3. RoBERTa ROC curves for each fold.

References

1. Málovics, G.; Csigéné, N.N.; Kraus, S. The role of corporate social responsibility in strong sustainability. *J. Socio-Econ.* **2008**, *37*, 907–918. [CrossRef]
2. Lodhia, S.K. The need for effective corporate social responsibility/sustainability regulation. In *Contemporary Issues in Sustainability Accounting, Assurance and Reporting*; Emerald Publishing Limited: Bingley, UK, 2012; pp. 139–152.
3. Ascioglu, A.; Gonzalez, J.; Zbib, L. Analysis of Sustainability Reports for Top 20 Companies in the S&P 500 Index. *J. Impact ESG Invest.* **2022**, *2*, 82–94.
4. Nations, U. Transforming Our World: The 2030 Agenda for Sustainable Development. Available online: https://sdgs.un.org/2030agenda (accessed on 22 November 2022).
5. Fonseca, L.M.; Domingues, J.P.; Dima, A.M. Mapping the Sustainable Development Goals Relationships. *Sustainability* **2020**, *12*, 3359. [CrossRef]
6. Bonina, C.; Koskinen, K.; Eaton, B.; Gawer, A. Digital platforms for development: Foundations and research agenda. *Inf. Syst. J.* **2021**, *31*, 869–902. [CrossRef]
7. Deniz, A.; Angin, M.; Angin, P. Understanding IMF Decision-Making with Sentiment Analysis. In Proceedings of the 2022 30th Signal Processing and Communications Applications Conference (SIU), Safranbolu, Turkey, 15–18 May 2022; pp. 1–4. [CrossRef]
8. Sovrano, F.; Palmirani, M.; Vitali, F. Deep Learning Based Multi-Label Text Classification of UNGA Resolutions. *CoRR* **2020**, abs/2004.03455.
9. Kim, N.; LaFleur, M. *What Does the United Nations "Say" about Global Agenda? An Exploration of Trends Using natUral Language Processing for Machine Learning*; DESA Working Paper No. 171; United Nations, Department of Economic and Social Affairs: New York, NY, USA, 2020.
10. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
11. Lee, J.S.; Hsiang, J. Patent classification by fine-tuning BERT language model. *World Pat. Inf.* **2020**, *61*, 101965. [CrossRef]
12. El-Alami, F.-Z.; Ouatik El Alaoui, S.; En Nahnahi, N. A multilingual offensive language detection method based on transfer learning from transformer fine-tuning model. *J. King Saud Univ.—Comput. Inf. Sci.* **2022**, *34*, 6048–6056. [CrossRef]
13. Khan, J.Y.; Khondaker, M.T.I.; Afroz, S.; Uddin, G.; Iqbal, A. A benchmark study of machine learning models for online fake news detection. *Mach. Learn. Appl.* **2021**, *4*, 100032. [CrossRef]
14. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv* **2019**, arXiv:1907.11692.
15. Casola, S.; Lauriola, I.; Lavelli, A. Pre-trained transformers: An empirical comparison. *Mach. Learn. Appl.* **2022**, *9*, 100334. [CrossRef]
16. Rodrawangpai, B.; Daungjaiboon, W. Improving text classification with transformers and layer normalization. *Mach. Learn. Appl.* **2022**, *10*, 100403. [CrossRef]
17. Briskilal, J.; Subalalitha, C. An ensemble model for classifying idioms and literal texts using BERT and RoBERTa. *Inf. Process. Manag.* **2022**, *59*, 102756. [CrossRef]
18. Yeh, C.; Meng, C.; Wang, S.; Driscoll, A.; Rozi, E.; Liu, P.; Lee, J.; Burke, M.; Lobell, D.B.; Ermon, S. SustainBench: Benchmarks for Monitoring the Sustainable Development Goals with Machine Learning. In Proceedings of the Thirty-Fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2), Virtual, 6–14 December 2021.
19. Matsui, T.; Suzuki, K.; Ando, K.; Kitai, Y.; Haga, C.; Masuhara, N.; Kawakubo, S. A natural language processing model for supporting sustainable development goals: Translating semantics, visualizing nexus, and connecting stakeholders. *Sustain. Sci.* **2022**, *17*, 969–985. [CrossRef] [PubMed]
20. Nilsson, M.; Chisholm, E.; Griggs, D.; Howden-Chapman, P.; McCollum, D.; Messerli, P.; Neumann, B.; Stevance, A.S.; Visbeck, M.; Stafford-Smith, M. Mapping interactions between the sustainable development goals: lessons learned and ways forward. *Sustain. Sci.* **2018**, *13*, 1489–1503. [CrossRef]
21. Smith, T.B.; Vacca, R.; Mantegazza, L.; Capua, I. Natural language processing and network analysis provide novel insights on policy and scientific discourse around Sustainable Development Goals. *Sci. Rep.* **2021**, *11*, 22427. [CrossRef]
22. Toetzke, M.; Banholzer, N.; Feuerriegel, S. Monitoring global development aid with machine learning. *Nat. Sustain.* **2022**, *5*, 533–541. [CrossRef]
23. Pukelis, L.; Bautista-Puig, N.; Skrynik, M.; Stanciauskas, V. OSDG—Open-Source Approach to Classify Text Data by UN Sustainable Development Goals (SDGs). *CoRR* **2020**, abs/2005.14569.
24. Amel-Zadeh, A.; Chen, M.; Mussalli, G.; Weinberg, M. NLP for SDGs: Measuring Corporate Alignment with the Sustainable Development Goals. *J. Impact ESG Invest.* **2022**, *2*, 61–81. [CrossRef]
25. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
26. Le, Q.; Mikolov, T. Distributed representations of sentences and documents. In Proceedings of the International Conference on Machine Learning, PMLR, Bejing, China, 22–24 June 2014; pp. 1188–1196.
27. Guisiano, J.; Chiky, R. Automatic classification of multilabel texts related to Sustainable Development Goals (SDGs). In Proceedings of the TECHENV EGC2021, Montpellier, France, 26 January 2021.

28. Guisiano, J.E.; Chiky, R.; de Mello, J. SDG-Meter: A deep learning based tool for automatic text classification of the Sustainable Development Goals. In Proceedings of the ACIIDS: 14th Asian Conference on Intelligent Information and Database Systems, Ho Chi Minh City, Vietnam, 28–30 November 2022.
29. Hajikhani, A.; Suominen, A. The interrelation of sustainable development goals in publications and patents: A machine learning approach. *CEUR Workshop Proc.* **2021**, *2871*, 183–193.
30. Natural Language Toolkit. Available online: https://www.nltk.org/ (accessed on 24 September 2022).
31. Miller, G.A. *WordNet: An Electronic Lexical Database*; MIT Press: Cambridge, MA, USA, 1998.
32. Ramos, J. Using tf-idf to determine word relevance in document queries. In Proceedings of the First Instructional Conference on Machine Learning, Los Angeles, CA, USA, 23–24 June 2003.
33. Hugging Face. Available online: https://huggingface.co/ (accessed on 24 September 2022).
34. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.U.; Polosukhin, I. Attention is All you Need. In *Proceedings of the Advances in Neural Information Processing Systems*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2017; Volume 30.
35. OSDG; UNDP IICPSD SDG AI Lab; PPMI. OSDG Community Dataset (OSDG-CD). 2022. Available online: https://zenodo.org/record/6393942#.Y4Q65X1BxPY (accessed on 24 September 2022).
36. Google. Colab. Available online: https://colab.research.google.com/ (accessed on 20 November 2022).