Knowledge Graph Data Enrichment Based on a Software Library for Text Mapping to the Sustainable Development Goals

Ioanna Mandilara[1], Eleni Fotopoulou[1], Christina Maria Androna[1], Anastasios Zafeiropoulos[1,*] and Symeon Papavassiliou[1]

[1]*Institute of Communication and Computer Systems, National Technical University of Athens, Athens, Greece*

Abstract

Over the last few years, there has been a significant increase in the release of massive amounts of data related to the Sustainable Development Goals (SDGs). This has been driven by the recognition that data is critical for monitoring progress towards the SDGs, identifying areas that require attention, and informing policy decisions. Given that a significant percentage of the provided information is made available in documents, the availability of software libraries that can enable scientists to easily extract information regarding the importance given to the SDGs in the documents' text is considered crucial. Towards this direction, we have developed an open-source Python software library that provides the mapping between text and the SDGs, taking advantage of novel machine learning techniques. The provided software library is modular and permits the dynamic selection of trained machine learning models for analysis purposes. The outcomes from the usage of the software library are fed for data population of a knowledge graph that is targeted to the tracking of information around the SDGs. The overall approach is open and can be easily adopted by scientists and policy makers to support participatory modeling processes, as well as participatory decision making and action planning for the development of solutions for climate-resilient regions.

Keywords

Knowledge Graph, Text classification, Sustainable Development Goals, Natural Language Processing, Keywords extraction

1. Introduction

The Sustainable Development Goals (SDGs) are a set of 17 interconnected global goals that aim to provide a universal framework for countries, organizations, and individuals to work together towards achieving a sustainable future for all [1]. They aim to address the most pressing global challenges, including poverty, inequality, climate change, environmental degradation, and social injustice.

TEXT2KG 2023: Second International Workshop on Knowledge Graph Generation from Text, May 28 - Jun 1, 2023, co-located with Extended Semantic Web Conference (ESWC), Hersonissos, Greece

*Corresponding author.

ioannamand@netmode.ntua.gr (I. Mandilara); efotopoulou@netmode.ntua.gr (E. Fotopoulou); andronaxm@netmode.ntua.gr (C. M. Androna); tzafeir@cn.ntua.gr (A. Zafeiropoulos); papavass@mail.ntua.gr (S. Papavassiliou)

0009-0005-1794-9700 (I. Mandilara); 0000-0001-7683-4616 (E. Fotopoulou); 0000-0003-0286-6564 (C. M. Androna); 0000-0003-0078-8697 (A. Zafeiropoulos); 0000-0002-9459-318X (S. Papavassiliou)

A lot of information exists around the SDGs, but much of it is scattered across multiple data silos and made available in different formats, making it difficult to gain a complete picture of progress. While quantitative data such as statistics and numerical indicators (e.g., in time series databases) are essential for measuring progress, qualitative data contained within text documents (e.g., international reports, policies, recommendations) is equally important for providing context and understanding the factors that contribute to progress or hinder it. To fully understand the progress being made towards the SDGs, it is therefore necessary to analyze and classify text documents based on their relevance to the SDGs.

To achieve so, novel machine learning techniques can be adopted such an Natural Language Processing (NLP) techniques. NLP techniques can help to overcome some of the challenges associated with manual analysis of text documents. For example, identifying the relevance of a particular policy or report to the SDGs can be a time-consuming and resource-intensive process, requiring a significant amount of manual effort. NLP techniques can automate this process, making it faster and more accurate, while also reducing the potential for human bias. By taking advantage of NLP techniques, it becomes possible to gain a more comprehensive and nuanced understanding of the factors that contribute to progress towards the SDGs, enabling policymakers and other stakeholders to make informed decisions and take targeted actions.

In the work presented in this manuscript, we provide details for the development of SDGDetector, an open-source software library that takes as input text and provides as output the association of the text description with the various SDGs. SDGDetector is based on two NLP techniques. It combines a traditional machine learning technique based on keywords detection and a deep learning technique that uses a transformer-based model. The produced mappings are fed as input for data population of the SustainGraph [2] that is an open-source Knowledge Graph (KG) that aims to track the progress achieved towards the targets defined for the various SDGs. In this way, we consider the fusion of information produced by the text analysis process with information that is made available in the SustainGraph. The latter includes time-series data for the various SDG indicators in global, regional and local level; time-series data for social, economic or environmental indicators; and data associated with the implementation of case studies focusing on the development of climate-resilient regions across Europe [3]. Over the information made available in the SustainGraph, analysis pipelines can be developed.

2. Text Analysis and Classification

2.1. Policies Overview

Overall, the policies around the SDGs are diverse and varied, reflecting the complex and interconnected nature of the challenges they seek to address and the need for the provision of automated text analysis tools to scientists and policy makers. The Paris Agreement is a legally binding international treaty on climate change adopted in 2015 by 196 parties [4]. Its goal is to limit global warming to well below 2 degrees Celsius above pre-industrial levels, with an aspiration to limit it to 1.5 degrees Celsius. To achieve this, countries pledge to nationally determine and communicate their own climate actions, known as Nationally Determined Contributions (NDCs), and to regularly report on their progress [5]. Such progress is in multiple cases associated with the SDGs, based on the 2030 Agenda for Sustainable Development that

sets out 17 goals, 169 targets, and 232 indicators, covering a range of economic, social, and environmental issues [1]. At the European Union (EU) level, the European Green Deal (EGD) is a comprehensive plan to transform the region into a sustainable, carbon-neutral economy by 2050. The EGD identifies several priority areas, while multiple documents are produced per year for specifying the action plan per priority area.

From an economic development perspective, the Country Specific Recommendations (CSRs) issued by the European Commission to individual EU member states aim to address a wide range of policy areas, including climate change. The CSRs relevant to climate change typically focus on increasing renewable energy sources, improving energy efficiency, promoting sustainable transport, and reducing greenhouse gas emissions in various sectors. Furthermore, the EU taxonomy is a classification system that defines environmentally sustainable economic activities and sets out criteria for determining whether an economic activity contributes to environmental objectives [6]. The EU taxonomy is closely linked to the EGD, as it aims to support the transition to a sustainable, low-carbon economy by providing a common language for investors, companies, and policymakers to identify and promote sustainable investments.

In our work in this manuscript, we focus on analysis of documents coming from the EGD and the CSRs with the usage of the SDGDetector. The produced mappings are introduced into the SustainGraph [2], where further information is made available and can be jointly analyzed. Such information regards the status of the SDG indicators in national and regional level, the mapping of the NDCs with the SDGs, as well as the classification of activities of main stakeholders in the case studies of the ARSINOE H2020 project [3] according to the EU taxonomy.

2.2. Overview of Natural Language Processing Tools focused to SDGs

Various NLP mechanisms are made available for examining the association between text documents and the SDGs. In [7], NLP methods are applied in combination with network analysis techniques to measure overlaps in international policy discourse around the SDGs. The produced results identify a strong discursive divide between environmental goals and all other SDGs, as well as the appearance of unexpected interdependencies between SDGs in different areas. In [8], a deep-learning natural language processing model in Japanese is applied based on bidirectional encoder representations from transformers (BERT) to support the mapping of text documents with the SDGs.

A set of NLP tools have been also made available to map text to the SDGs, such as SDG-meter [9], text2sdg R package [10], OSDG-ai [11], LinkedSDG [12], SDG-tracker [13], SDGMapper [14] and SDG Pathfinder [15]. SDG-meter is proposed as an open-source online tool able to indicate the SDGs linked to an input text, taking advantage of a multi-label classification of texts using BERT [9] and allowing users to compare the accuracy of different mapping algorithms. The text2sdg R library [10] open-source package detects SDGs in text data using different existing or custom-made query systems, while the OSDG-ai [11] and LinkedSDG [12] tools take advantage of ontologies and keywords matching techniques. SDG-tracker [13], SDGMapper [14] and SDG Pathfinder [15] regard online platforms or tools managed by different organizations that offer SDG mapping services.

Given the existence of such tools, the main motivation for the development of SDGDetector stems for the need to provide an open-source software library that can be easily accessible

and extensible by software developers. SDGDetector also provides easy parameterization capabilities and options for selection of the methods (keyword extraction and text classification) to be applied for the text analysis processes. SDGDetector is developed in Python and can be easily integratable in Python-based workflows that are used in data population processes of knowledge graphs.

3. Approach for mapping text to SDGs

3.1. Methodology

We propose two different NLP techniques to interlink text with the SDGs. The main idea is to combine a traditional machine learning technique, which uses keywords to find the relevance of texts with the SDGs, and a deep learning technique, namely transfer learning, which is based on a transformer-based model. In the first case, the linkage between the texts and the SDGs can be made by computing the cosine similarity scores between the text's keywords and the SDG's keywords. In the second case, the linkage can be expressed as the probability that the texts are related to the SDGs by using a transformer-based classifier. In the upcoming subsections we provide details for both techniques, while both of them are made openly available in a GitLab repository [16].

3.1.1. Text mapping to SDGs based on Keywords Extraction

To support text mapping to the SDGs based on keywords extraction, we have adopted a taxonomy of keywords per SDG, as it is provided by the Committee on the Environment, Climate Change, and Sustainability of the University of Toronto [17] for the SDGs from 1 to 16, as well as the taxonomy provided by the Monash University and the Sustainable Development Solutions Network (SDSN) for Australia, New Zealand and Pacific area for the SDG 17 [18]. Based on these taxonomies, we examine the matching among the extracted keywords from the applied process with the already classified keywords per SDG. The overall process is depicted in Figure 1.

The keywords extraction process consists of the following steps:

Data cleaning: The data quality of the text is improved by removing digits and special characters from the given text.

Candidate keywords extraction: The candidate n-gram words and/or phrases in the text are extracted by using the *bag-of-n-grams* representation [19]. This representation maps a text document as an unordered collection of its n-grams and is able to eliminate stop words and tokenize plain texts.

Embeddings: The candidate keywords/key phrases and the entire document are converted into numerical data using embeddings. For this purpose, sentence transformers are used based on the pre-trained *all-mpnet-base-v2* sentence transformer model [20], which has shown high performance for sentence embeddings and semantic search and it is recommended from the python library *sentence-transformers*.

Representative keywords identification: The most representative keywords/ key phrases of the text are extracted based on a cosine similarity score. The Maximal Margin Relevance (MMR) algorithm [21] is applied to generate keywords/key phrases based on cosine similarity.

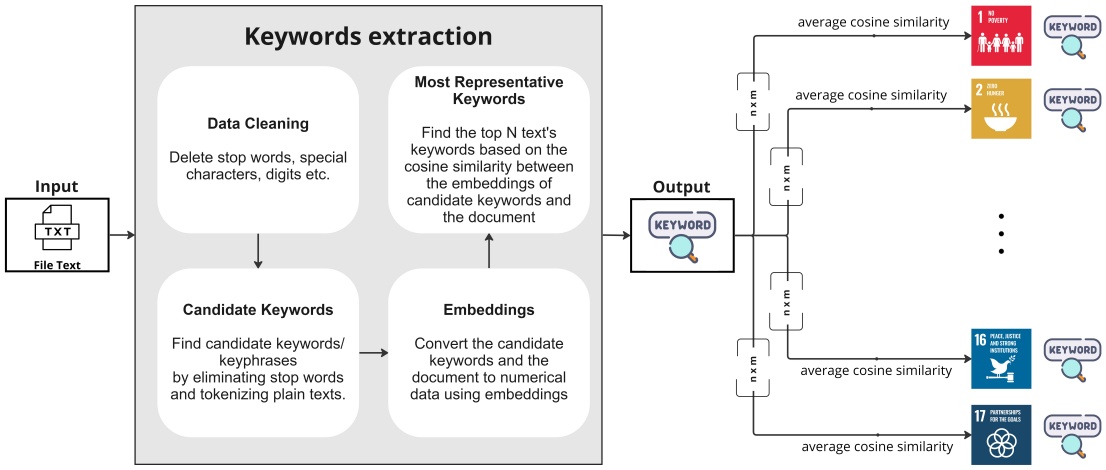

Figure 1: Keywords extraction process.

The MMR attempts to decrease the redundancy while increasing the diversity of outputs. The keywords or key phrases that most closely match the document are first chosen. Then, we iteratively choose new candidates that are both similar to the document and not identical to the previously selected keywords/keywords.

By having extracted the representative keywords from the text, the cosine similarity matrix is computed between their embeddings and the embeddings of the classified keywords per SDG. The cosine similarity matrix is an (n x m) matrix, where n is the number of the top exemplary keywords and m is the number of the SDG's keywords. The higher the value of the average cosine similarity between two terms, the greater their relevance.

3.1.2. Multi-label Classification using pre-trained Transformer-based Models

A deep learning technique has been developed to find the similarities between the texts and the SDGs, considering the tackled problem as a multi-label classification problem. The overall classification process is depicted in Figure 2.

The training and validation datasets for the model are based on the *OSDG Community Dataset*[22]. This dataset is made up of paragraph-length text samples obtained from publicly available publications such as reports, policy documents, and publication abstracts. It consists of 37575 text excerpts and includes the related SDG, the number of volunteers who voted against the proposed SDG label (labels_negative), the number of volunteers who voted in favor of the proposed SDG label (labels_positive), and the agreement score based on the formula:

$$agreement = \frac{|labels_{positive} - labels_{negative}|}{labels_{positive} + labels_{negative}} \quad (1)$$

The text classification process consists of the following steps:
Data preparation: Only data with an accepted SDG label (labels_positive > labels_negative) and a score of agreement greater than 0.55 were chosen for the current study. The decision

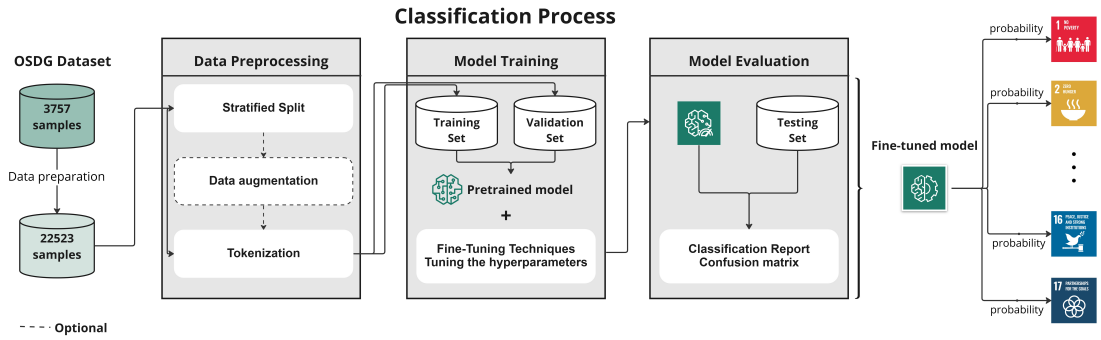

Figure 2: Classification process.

to only include samples with an agreement over 0.55 aims to ensure reliable and consistent labeling of text data by minimizing subjective interpretation among volunteers. In this manner, 22523 out of 37575 samples were retained. Figure 3 presents the number of text excerpts used for the SDGs from 1 to 16 upon the filtering process.

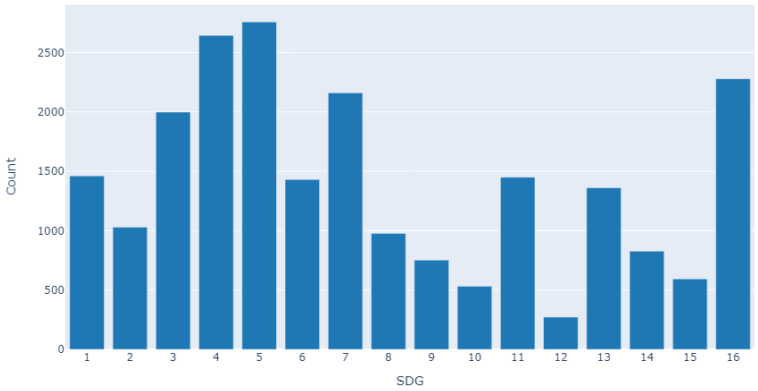

Figure 3: Distribution of text excerpts (agreement > 0.55) over the SDGs.

Data preprocessing: The data preprocessing phase involves the splitting of the dataset into training, testing, and validation sets and the tokenization of each set to ensure it meets the expected format for the pre-trained models. We have first splitted the dataset into training and testing sets, by using the 80% and 20% of the samples, respectively. Following, we have further divided the training set into training and validation sets, with 80% and 20% of the training samples, respectively. The validation set is reserved for hyperparameter optimization and performance evaluation. To handle any dataset imbalance, we have applied a *stratified* split that maintained the same class distribution percentages in each split.

Moreover, we have used data augmentation techniques to generate synthetic training data for classes containing fewer than 1000 texts. This process involved drawing a random sample from

each minority class and sequentially introducing it into four augmenters in a data augmentation pipeline. The pipeline starts by injecting new words into a random position(*insertion*) based on the BERT model's contextual word embeddings calculation and then replacing different words by their contextual embeddings (*substitution*). Next, we replaced the text's words using WordNet synonyms (*synonym replacement*) and added additional text (*sentence augmentation*) based on the XLNet model's contextual word embeddings.

To prepare the textual data for the deep learning model, we used appropriate tokenization based on the model being trained, which converted the unstructured text strings into a numerical data structure.

Model training: At this phase, two different fine-tuning techniques are used:

- **Train the entire architecture**: In this case, we train the model by unfreezing all the layers of its architecture, i.e. the pre-trained weights of the model are updated based on the new dataset, namely the OSDG Community Dataset.
- **Train some layers while freezing others**: In this case, we train the model partially by freezing its initial layers and training only the last ones with the new dataset.

We experiment with five different transformer-based language models by using one or both of the fine-tuning techniques. Namely, we consider the base version of BERT (Bidirectional Encoder Representations from Transformers) [23] that is trained on large amounts of text data and can be fine-tuned for a variety of NLP tasks; the base version of RoBERTa (Robustly Optimized BERT Pretraining Approach) model that is a modified version of BERT [24], uses a larger amount of training data and a longer pre-training process to achieve better performance; the XLNet (eXtreme MultiLingual pretraining for language understanding) model [25] that uses an autoregressive pre-training method and is designed to improve on the limitations of BERT and other transformer-based models; the GPT-2 (Generative Pre-trained Transformer 2) model that uses the transformer architecture like BERT, but the decoder part instead of the encoder part; and the GPT-Neo (Generative Pre-trained Transformer-Neo) model that is a community-driven effort to replicate the success of GPT models with a focus on open-source and democratizing access to large-scale NLP models.

Different NLP tasks can be accomplished using the aforementioned models. For the classification task, we add an extra layer of untrained linear neurons on top of these models. During training, these neurons are updated to map the samples to one of the 16 classes. Afterwards, optimizing of the hyper-parameters is crucial for both fine-tuning techniques. Optimal values are selected for the batch size (number of training samples), the max sequence length (maximum length of the sequence in tokens), the optimizer learning rate, and the number of epochs.

Model evaluation: The final stage of the classification process regards the evaluation part that is based on different metrics to understand the fine-tuned model's performance, as well as its strengths and weaknesses. The classification report, evaluation plots, and confusion matrix of the testing set are commonly used for this purpose.

The *classification report* is used to measure the quality of predictions. The report shows the main classification metrics precision, recall, and f1-score on a per-class basis and in total. For our classification problem, the metric f1-score is considered more suitable than the accuracy, since the dataset is imbalanced. The *confusion matrix* is a 16 x 16 matrix that compares the actual and predicted values, providing a tabular way of visualizing the model performance. The

evaluation plots show the loss/ f1-score of the training and the validation samples during the training.

3.1.3. Combinatory evaluation index

Based on the outcomes provided by the aforementioned techniques, we consider the evaluation of an index that combines both of them. The index is called as r_{SDG} index and is based on the formula:

$$r_{SDG} = \begin{cases} 0.7 * probability + 0.3 * avg\_cosine\_similarity & \text{, for SDGs 1-16} \\ 0.5 * avg\_cosine\_similarity & \text{, for SDG 17} \end{cases} \quad (2)$$

The r_{SDG} index has two branches since there is no available text classification dataset for the SDG 17. Thus, for this case, the evaluation is based exclusively on the applied keywords matching technique. The probability coefficient exhibits a higher value than the corresponding coefficient of the average cosine similarity, as the models used in the text classification technique are able to better capture the complex relationships between words in a sentence.

3.2. SDGDetector Software Library

In this section, we present *SDGDetector*, an open-source Python library that we have developed that streamlines the process of mapping textual data to the Sustainable Development Goals (SDGs). SDGDetector is made openly available in a GitLab repository [26]. The library consists of three primary classes that provide a powerful set of tools for automated SDG classification, as detailed in Table 1.

Table 1
Overview of SDGDetector classes.

Class	Description
SDG_classifier_using_model	Using a pre-trained fine-tuned XLNet or RoBERTa model, it determines the probability that the given text belongs to the SDGs.
SDG_classifier_using_keywords_extraction	Using pre-trained sentence-transformer models to extract the most significant terms of the provided texts and compute the similarity scores of them with the SDG's keywords.
SDG_classifier	It returns the r_{SDG} index, by combining the aforementioned classes.

The class *SDG_classifier_using_model* leverages a fine-tuned pre-trained XLNet or RoBERTa model developed in Python using Pytorch. It includes a *predict* method that takes a list of texts as input and returns the most pertinent SDG, its corresponding name, and the probabilities of each text belonging to the first 16 SDGs. This class is flexible enough to work with a user's fine-tuned model or our pre-trained XLNet or RoBERTa model with a high f1-score of 0.90.

The class *SDG_classifier_using_keywords_extraction* incorporates the method delineated in Section 3.1.1. This class offers two methods, the *find_top_keywords* and the *predict* methods.

The *find_top_keywords* method extracts the top n keywords from the input texts using the embeddings generated by the sentence transformer model. This library features the Mpnet-base, MiniLM: 6 Layer Version, and the DistilBert-base sentence transformer models. The first two models yield the highest quality of embeddings, where the MiniLM: 6 Layer Version being 5 times faster than the Mpnet-base, while the latter offers the best quality of embeddings. The user can select the model to be used, as well as the diversity and range of keywords. The *predict* method returns the average cosine similarity with the SDGs.

The class *SDG_classifier* combines the previous methods and returns the r_{SDG} index for the given texts.

4. Evaluation Results

In this section, we provide a set of results based on the application of the developed approach and software library for the classification of texts coming from the *Country Specific Recommendations* and *European Green Deal Policies*, and the data population of SustainGraph based on the provided mappings.

4.1. Keywords Extraction and Similarity Score

Regarding the traditional machine learning method, the top 10 and 5 representative keywords for each section of the European Green Deal strategies and the Country specific Recommendations were produced respectively. For the sentence embeddings, the model All-Mpnet-Base-v2 was used, and the divergence of the algorithm MMR was set to 0.3.

The cosine similarity matrix was computed between the top 10 keywords of each section of the European Green Deal (EGD) strategies and the SDG's keywords. Similarity scores of less than 30% were not taken into consideration, similarity scores in the range of 30% to 50% were considered as medium, while similarity scores of more than 50% were considered as high. By taking the percentage of high and medium similarity scores for each EGD strategy and SDG respectively, we were able to get an overview of the association between them, as depicted in Figure 4. Depending on the EGD strategy, the most relevant SDG is identified with high score (e.g., for the Solar Energy EGD strategy, the SDG #7 (Clean and affordable energy) is dominant). The smaller association values (*yellow bars*) are shown for the SDGs #1 (No poverty), #10 (Reduce Inequalities), and #16 (Promote just, peaceful, and inclusive societies).

4.2. Transformer-Based model Fine-Tuning and Inference

Following, we conducted several experiments between the transformer-based models by using the OSDG community dataset, as mentioned in Section 3.1.2. The experiments were carried out using the Kaggle notebook's GPU hardware accelerator platform. Additionally, we used the Huggin Face's Transformers library as the source for all the transformer-based models, implemented in PyTorch.

Throughout our experiments, we used the Adam optimizer with weight decay fix [27] and a linear scheduler for 10% of the total steps. For each model's training, we selected the best fine-tuning learning rates (among 5e-5, 3e-5, and 2e-5), as suggested in [23]. In the case of the

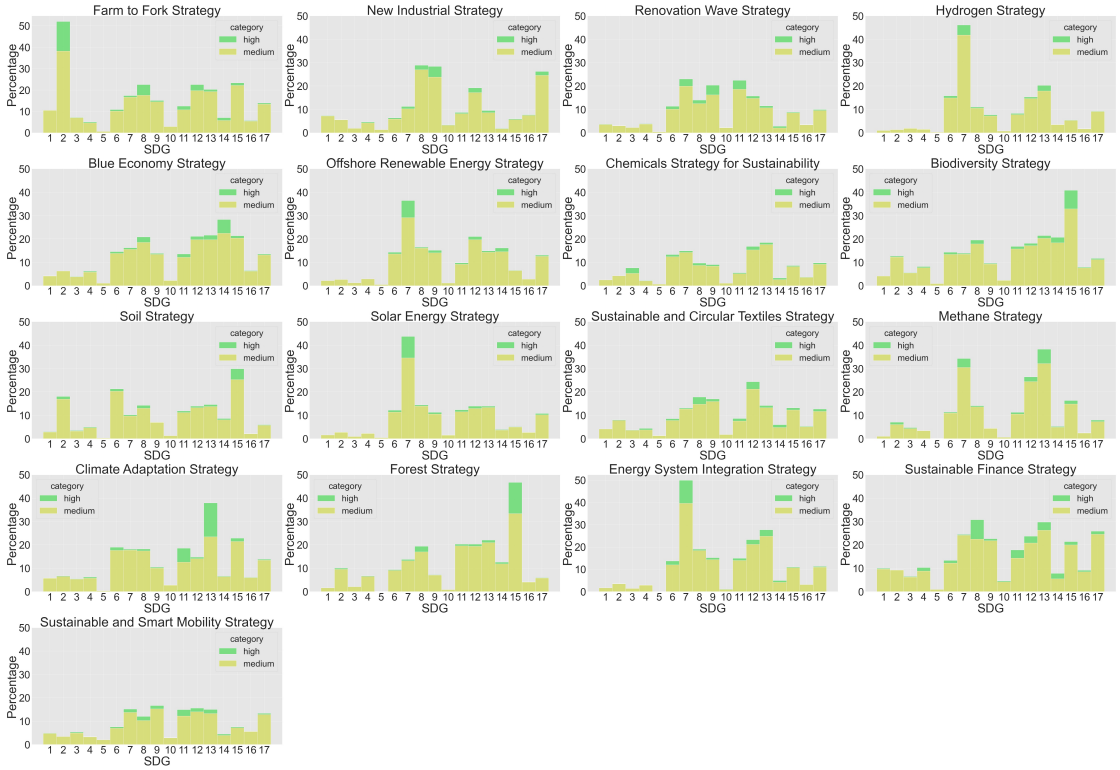

Figure 4: Percentage of high and medium similarity scores across the EGD Strategies.

Bert, RoBERTa, and XLNet models, the batch size and maximum length were set to 32 and 512, respectively. These models were trained for 3, 4, and 5 epochs. Since GPT models have higher memory requirements than the other models, we experimented with various combinations of maximum length and batch size to prevent memory issues.

As for the loss function, we optimized the Bert and RoBERTa models using the Cross-Entropy loss function, while the XLNet model was optimized with the BCE-with-logits loss function. In addition, we investigated the second fine-tuning approach for Bert, RoBERTa, and XLNet models, which involves the partial training of the model as discussed in Section 3.1.2, alongside the first fine-tuning technique, i.e. the training of the complete model, for the GPT models. In more detail, these 3 models consist of 12 layers with an added single linear layer on top, acting as the classifier. In our experiments, the pre-trained model parameters in the 1st to 11th layer were frozen, while the last 12th and the classifier layer were set as trainable.

4.2.1. Model training and evaluation

The Bert, RoBERTa, and XLNet models showed their best performance after 3 fine-tuning epochs and learning 5e-5, as demonstrated in Figure 5, while lower performance was noticed in case of the GPT2 and GPTNeo models. The three models (Bert, RoBERTa, XLNet) achieved similar f1-scores of approximately 0.90. The Bert model yielded the highest f1-score of 0.91,

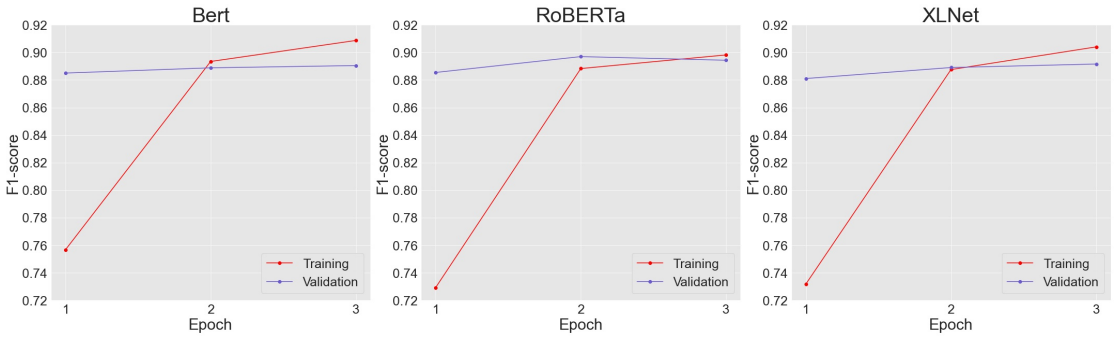

Figure 5: F1-score between Bert, RoBERTa, and XLNet models.

but the RoBERTa model had a slightly lower f1-score of 0.90, with less divergence between the evaluation and training curves. The XLNet model also achieved an f1-score of 0.90 with a smaller divergence between the training and evaluation curves than the Bert model. Therefore, we concluded that the RoBERTa model outperforms the Bert model and achieves comparable outcomes to the XLNet model. In 6, the confusion matrices are presented that illustrate the robust performance of the models, where all models yielded an f1-score of 0.90 over the testing set.

Furthermore, the data augmentation technique, described in Section 3.1.2, was used for the Bert, RoBERTa, and XLNet models, but all the models achieved training/validation/testing f1-scores of around 0.90, similar to the results obtained without it. Thus, we concluded that these models could handle imbalanced datasets.

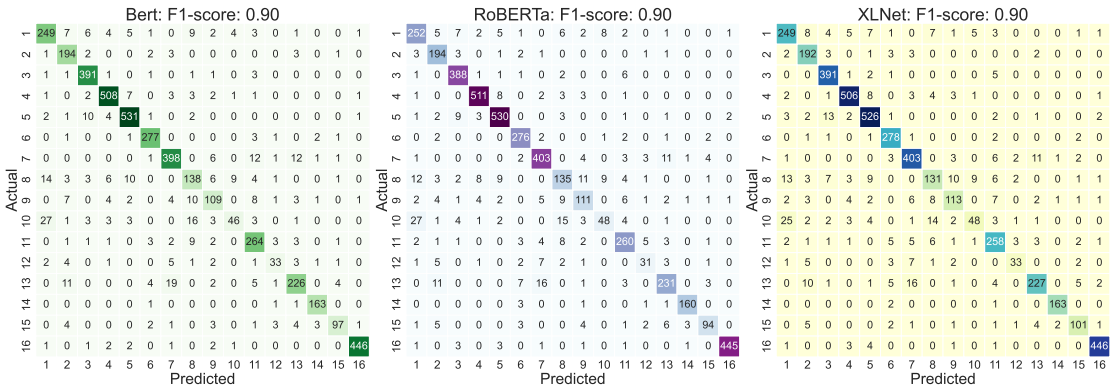

Figure 6: Confusion matrices between Bert, RoBERTa, and XLNet models.

4.2.2. Model inference

Following the training of the models, we fed the recommendations of each country and the sections of each European Green Deal Strategy into our fine-tuned XLNet model to create predictions for the first 16 SDGs. The sigmoid function is applied to each raw output value. We

considered probabilities greater than or equal to 10%, as probabilities less than that indicate shaky connections between the texts and the SDGs. Figure 7 displays the average probability per Goal for each EGD Strategy, with the majority of the strategies being linked to Goals #7 (Affordable and Clean Energy) and #13 (Climate Action). Notably, Goals #1 (No poverty), #10 (Reduce Inequalities) and #16 (Promote just, peaceful, and inclusive societies) are not represented in the EGD Strategies. As mentioned previously, the same SDGs are associated with only medium similarity scores with the EGD Strategies. These two approaches arrive at comparable conclusions, strengthening one another. Combining these two techniques may result in a more robust measure of association.

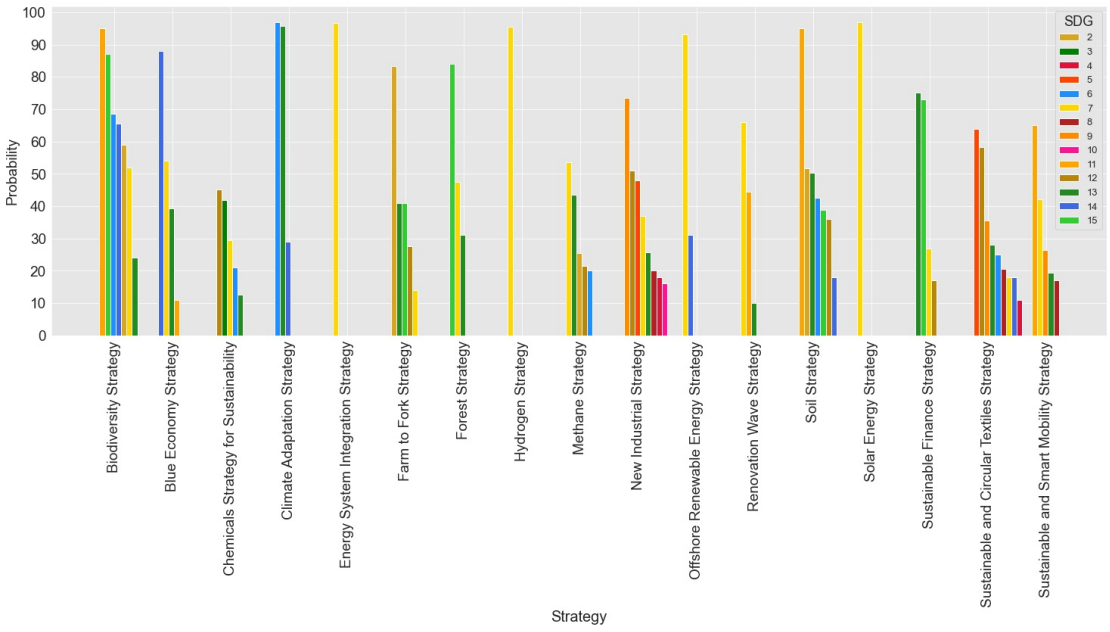

Figure 7: Average probability of each EGD Strategy with the SDGs.

4.3. Evaluation based on the r_{SDG} index

Following, we compared the effectiveness of the two NLP approaches for a text classification problem related to the SDGs. According to the *SDGDetector*, we may combine these two methodologies to discover the relationship between EGD strategies and CSRs with the SDGs. The method *predict* of the class *SDG_classifier* returns the r_{SDG} index, as explained in Section 3.1.3. The parameters used in the *predict* method are shown in Table 2, where the number of keywords extracted was chosen based on the text's length.

The plots presented in Figure 8 illustrate the r_{SDG} values for each EGD Strategy with respect to the SDGs. The results show that the Goal #7 (Affordable and Clean Energy) has the strongest connection with the Renovation Wave Strategy, Hydrogen Strategy, Solar Energy Strategy, Offshore Renewable Energy Strategy, Solar Energy Strategy, Methane Strategy, and Energy System Integration Strategy. In contrast, the Goal #13 (Climate Action) is mostly related to the

Table 2
Parameters of the predict method.

Parameters	EGD Strategies	CSRs
Transformer-based model	XLNet	XLNet
Sentence-transformer model	All mpnet base v2	All mpnet base v2
Top keywords	10	5
Diversity	0.3	0.3
Range of keywords	(1,2)	(1,2)

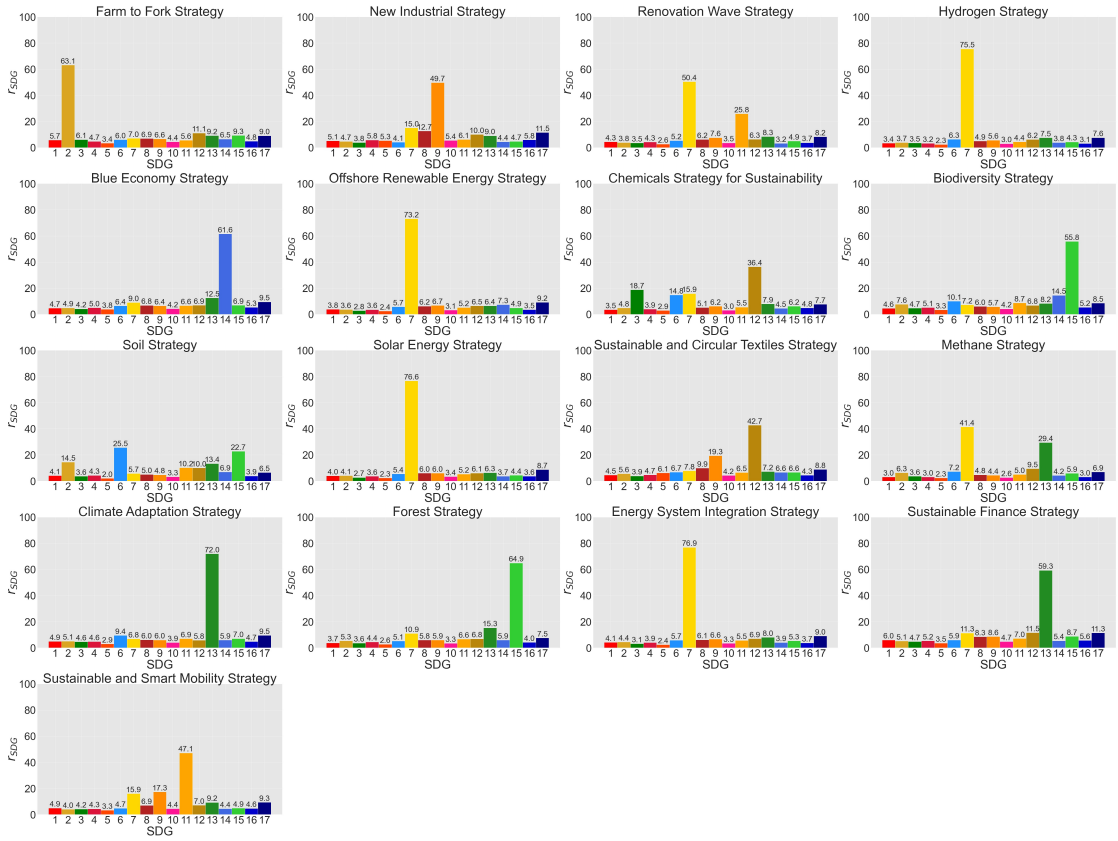

Figure 8: Average r_{SDG} of each EGD Strategy with the SDGs.

Climate Adaptation Strategy and Sustainable Finance Strategy. The Forest and Biodiversity Strategy is highly associated with Goal #15 (Life on Land).

To validate the outcomes provided by the SDGDetector with alternative tools that exist, a comparison has been made with the outcomes provided by the SDGMapper [14] tool, that is developed by the European Commission under the KnowSDGs web platform. The SDGMapper tool expresses the linkage between a document and an SDG as the ratio of keywords in each goal to the total number of keywords detected. By uploading the EGD Strategies into this tool we are able to compare our findings with those from a tool provided by the European Commission.

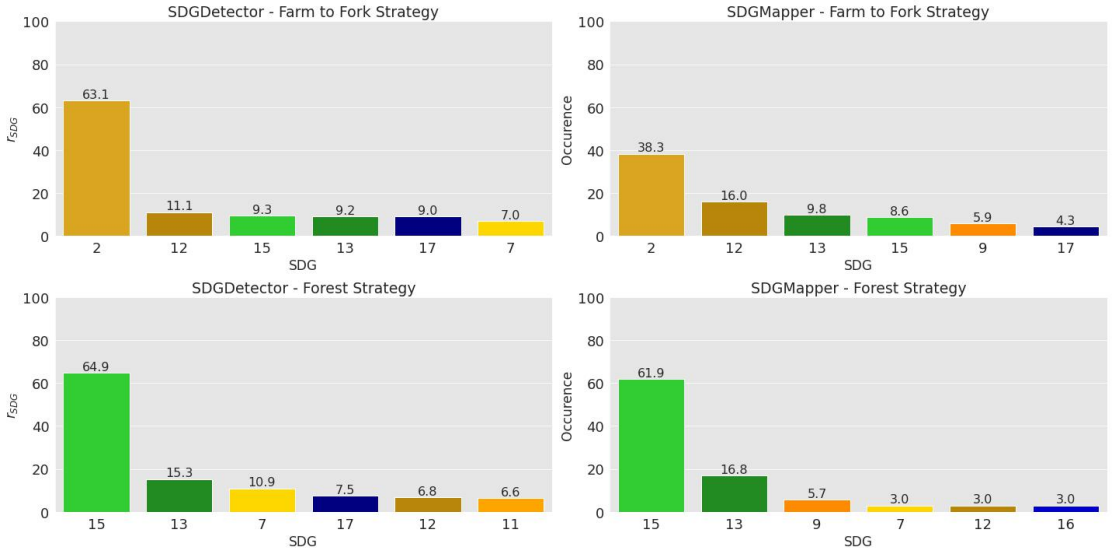

Figure 9: Comparison between SDGDetector and SDGMapper.

Figure 9 shows the top 6 related SDGs in two indicative EGD Strategies as identified by the SDGMapper tool and our Python library SDGDetector. For the Farm to Fork strategy, both tools identified the Goals #2 (Zero Hunger) and #12 (Responsible Consumption and Production) as the most relevant SDGs, while high association is present in the rest Goals. For the Forest Strategy, the top two SDGs were the same in both tools (#15 (Life on Land) and #13 (Climate Action)), while there is also good mapping in the rest Goals that present much smaller probabilities.

5. SustainGraph Enrichment

SustainGraph is a Knowledge Graph (KG) that tracks information related to the progress towards the achievement of targets defined in the SDGs at national and regional levels, as well as further social, economic and environmental indicators that may be proven useful to inter-disciplinary scientists in their modeling and analysis processes [2]. The objective is to serve as an open and comprehensive knowledge source for information related to the SDGs, utilizing graph databases and NLP techniques for data analysis and population. It is developed as a labeled property graph based on the Neo4j technology and is made available in a GitLab repository [28].

The overall structure of SustainGraph is depicted in Figure 10. At the left part of the figure, various policies frameworks are listed, where information coming from policies documents, strategies and directives is introduced. The SDGDetector library is used for the development of data population pipelines for automating the population of the SustainGraph with data coming from the strategies defined in the EGD and the CSRs per country. Such information can be combined with data coming from the tracking of the SDG targets and indicators, data coming from third-party sources, as well as data coming from the implementation of cases studies. A closer view on the structure of the conceptualization of the SustainGraph regarding the EGD

and CSR entities is made available in Figure 11. For both EGD and CSR documents, we keep track of the year that they are issued and their association with the SDGs, while for the CSRs we add information regarding the country (geoarea) that they are issued for.

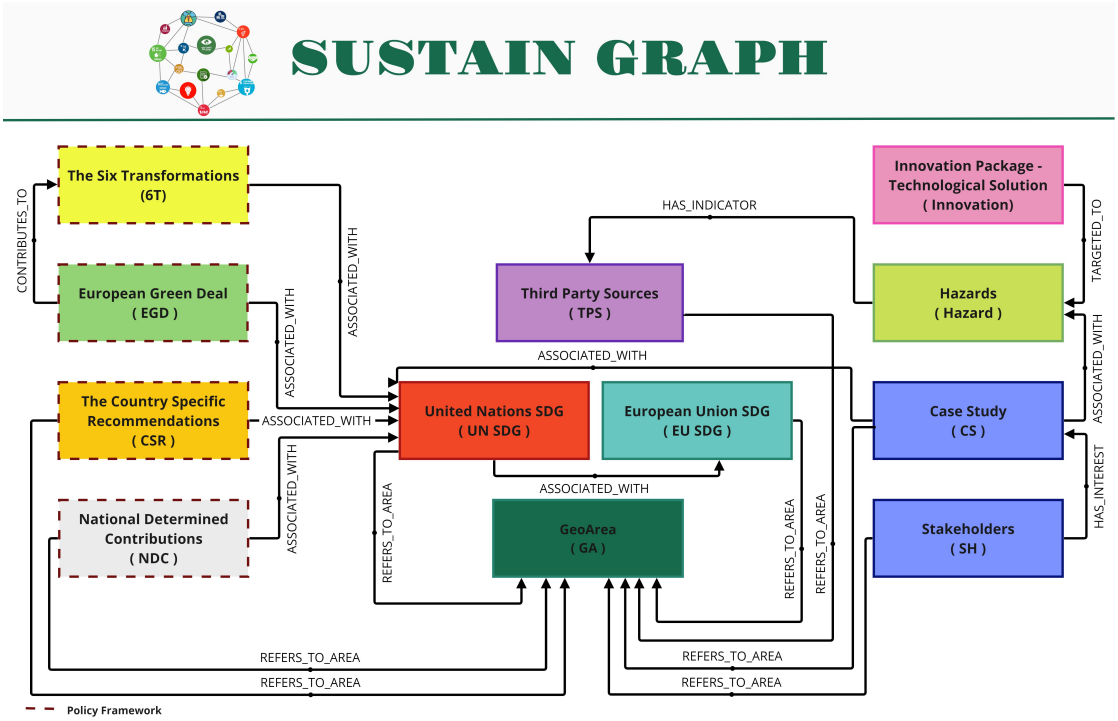

Figure 10: Overview of the SustainGraph.

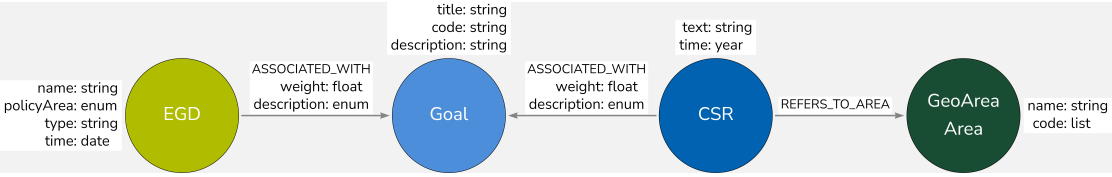

Figure 11: Policy Documents in the SustainGraph.

The association is based on the *weight* and the *description* properties. The *weight* regards the exact association value, as provided by the r_{SDG} index of SDGDetector, while the *description* is produced based on a rule-based labeling approach. Relationships with weights less than 10% suggest poor linkages between policies and the SDGs and are classified as *very low*, whereas those with values in the range $[10, 30)$ indicate a more powerful but insufficient linkage and are classified as *low*. Additionally, the relationships with weight in the range $[30, 60)$ are classified as *medium* and the relationships with weights in the range $[60, 100)$ are classified as high.

Based on the data population of the SustainGraph, indicative visualisations are produced. The pie chart in Figure 12 illustrates the association of the CSRs with the SDGs in the years 2011

and 2022, that correspond to the first year where the CSRs were issued and the year where their latest version was made available. The chart shows the percentage of CSRs that are associated with high and medium weights in the relationship "ASSOCIATED_WITH" with the SDGs, per Goal. It can be claimed that Europe's focus in 2011 was primarily on the areas of economic growth and educational quality (Goals 8 and 4), while in 2022, the majority of recommendations are targeted at the areas of clean energy and climate action (Goals 7 and 13).

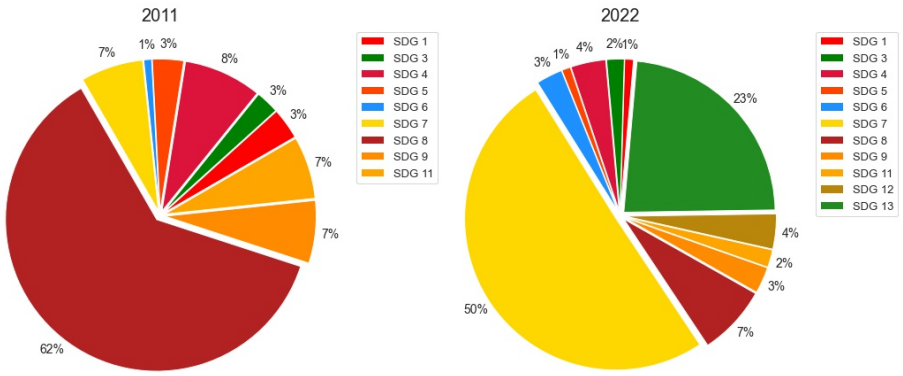

Figure 12: The percentage of CSRs with high or medium weights with the SDGs across time.

6. Conclusions and Future Work

Upon highlighting the need for the provision of open-source tools to analyze the relevance of documents with the SDGs, SDGDetector is detailed as an open-source software library in Python that supports such an analysis. SDGDetector is modular, easily adoptable and extensible by software developers, while it permits the selection of various models for doing the analysis. A composite index is produced for the analysis results, combining input coming from a traditional machine learning technique based on keywords matching and a deep learning technique that takes advantage of transformer-based models. The produced outcome is used for data population of an open-source Knowledge Graph for tracking the progress towards the achievement of the SDGs. A set of evaluation results are made available based on analysis of documents coming from the CSRs and strategies from the EGD, showcasing the suitability of the proposed approach for the identification of the association between the text and the SDGs. Based on the presented work, future research and development areas are identified. These include the support of participatory analysis processes based on the heterogeneous data made available in the SustainGraph, and the performance evaluation with models like GPT, given that access to a bigger infrastructure with GPU (graphics processing unit) support can be provided.

Acknowledgments

This research work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101037424.

References

[1] B. Lee, F. Kjaerulf, S. Turner, L. Cohen, P. Donnelly, R. Muggah, R. Davis, A. Realini, B. Kieselbach, L. Snyder, I. Waller, R. Gordon, M. Moloney-Kitts, G. Lee, J. Gilligan, Transforming our world: Implementing the 2030 agenda through sustainable development goal indicators, Journal of public health policy 37 (2016) 13–31. doi:10.1057/s41271-016-0002-7.

[2] E. Fotopoulou, I. Mandilara, A. Zafeiropoulos, C. Laspidou, G. Adamos, P. Koundouri, S. Papavassiliou, Sustaingraph: A knowledge graph for tracking the progress and the interlinking among the sustainable development goals' targets, Frontiers in Environmental Science 10 (2022). URL: https://www.frontiersin.org/articles/10.3389/fenvs.2022.1003599. doi:10.3389/fenvs.2022.1003599.

[3] ARSINOE project, ARSINOE H2020 project: Climate Resilient Regions Through Systemic Solutions and Innovations. (2023). Available at https://arsinoe-project.eu/.

[4] C. A. Horowitz, Paris agreement, International Legal Materials 55 (2016) 740–755. doi:10.1017/S0020782900004253.

[5] United Nations, Climate Action, United Nations, All About the NDCs. (2023). Available at https://www.un.org/en/climatechange/all-about-ndcs.

[6] J. Dusík, A. Bond, Environmental assessments and sustainable finance frameworks: will the eu taxonomy change the mindset over the contribution of eia to sustainable development?, Impact Assessment and Project Appraisal 40 (2022) 90–98. URL: https://doi.org/10.1080/14615517.2022.2027609. doi:10.1080/14615517.2022.2027609. arXiv:https://doi.org/10.1080/14615517.2022.2027609.

[7] T. B. Smith, R. Vacca, L. Mantegazza, I. Capua, Natural language processing and network analysis provide novel insights on policy and scientific discourse around Sustainable Development Goals, Scientific Reports 11 (2021) 22427. URL: https://www.nature.com/articles/s41598-021-01801-6. doi:10.1038/s41598-021-01801-6, number: 1 Publisher: Nature Publishing Group.

[8] T. Matsui, K. Suzuki, K. Ando, Y. Kitai, C. Haga, N. Masuhara, S. Kawakubo, A natural language processing model for supporting sustainable development goals: translating semantics, visualizing nexus, and connecting stakeholders, Sustainability Science 17 (2022) 969–985. URL: https://doi.org/10.1007/s11625-022-01093-3. doi:10.1007/s11625-022-01093-3.

[9] J. E. Guisiano, R. Chiky, J. de Mello, SDG-Meter : a deep learning based tool for automatic text classification of the Sustainable Development Goals, in: ACIIDS :14th Asian Conference on Intelligent Information and Database Systems, Ho Chi Minh, Vietnam, 2022. URL: https://hal.science/hal-03738404.

[10] D. S. Meier, R. Mata, D. U. Wulff, text2sdg: An r package to monitor sustainable development goals from text, 2021. URL: https://arxiv.org/abs/2110.05856. doi:10.48550/ARXIV.2110.05856.

[11] L. Pukelis, N. Bautista-Puig, G. Statulevičiūtė, V. Stančiauskas, G. Dikmener, D. Akylbekova, Osdg 2.0: a multilingual tool for classifying text data by un sustainable development goals (sdgs), 2022. URL: https://arxiv.org/abs/2211.11252. doi:10.48550/ARXIV.2211.11252.

[12] A. Joshi, L. G. Morales, S. Klarman, A. Stellato, A. Helton, S. Lovell, A. Haczek, A knowledge

organization system for the united nations sustainable development goals, in: R. Verborgh, K. Hose, H. Paulheim, P.-A. Champin, M. Maleshkova, O. Corcho, P. Ristoski, M. Alam (Eds.), The Semantic Web, Springer International Publishing, Cham, 2021, pp. 548–564.
[13] Ritchie, Roser, Mispy, Ortiz-Ospina., Measuring progress towards the Sustainable Development Goals. (2023). Available at https://sdg-tracker.org/.
[14] European Commission, SDG Mapper. (2023). Available at https://knowsdgs.jrc.ec.europa.eu/sdgmapper.
[15] Organization for Economic Cooperation and Development, SDG Pathfinder. (2023). Available at https://sdg-pathfinder.org/.
[16] I. Mandilara, E. Fotopoulou, C.M Androna, A. Zafeiropoulos, SustainNLP Gitlab Repository. (2023). Available at https://gitlab.com/netmode/sdg-text2kg.
[17] Committee on the Environment, Climate Change, and Sustainability, University of Toronto., Sustainable Development Goals (SDGs) Keywords. (2023). Available at https://sustainability.utoronto.ca/inventories/sustainable-development-goals-sdgs-keywords/.
[18] SDSN Australia, New Zealand and Pacific area., Sustainable Development Goals (SDGs) Keywords. (2023). Available at https://ap-unsdsn.org/regional-initiatives/universities-sdgs/.
[19] K. Juluru, H.-H. Shih, K. Murthy, P. Elnajjar, Bag-of-words technique in natural language processing: A primer for radiologists, RadioGraphics 41 (2021) 210025. doi:10.1148/rg.2021210025.
[20] K. Song, X. Tan, T. Qin, J. Lu, T. Liu, Mpnet: Masked and permuted pre-training for language understanding, CoRR abs/2004.09297 (2020). URL: https://arxiv.org/abs/2004.09297. arXiv:2004.09297.
[21] K. Bennani-Smires, C. Musat, A. Hossmann, M. Baeriswyl, M. Jaggi, Simple unsupervised keyphrase extraction using sentence embeddings, in: Proceedings of the 22nd Conference on Computational Natural Language Learning, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 221–229. URL: https://aclanthology.org/K18-1022. doi:10.18653/v1/K18-1022.
[22] OSDG, U. I. S. A. Lab, PPMI, Osdg community dataset (osdg-cd), 2022. URL: https://doi.org/10.5281/zenodo.7136826. doi:10.5281/zenodo.7136826.
[23] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, ArXiv abs/1810.04805 (2019).
[24] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized BERT pretraining approach, CoRR abs/1907.11692 (2019). URL: http://arxiv.org/abs/1907.11692. arXiv:1907.11692.
[25] Z. Yang, Z. Dai, Y. Yang, J. G. Carbonell, R. Salakhutdinov, Q. V. Le, Xlnet: Generalized autoregressive pretraining for language understanding, CoRR abs/1906.08237 (2019). URL: http://arxiv.org/abs/1906.08237. arXiv:1906.08237.
[26] I. Mandilara, E. Fotopoulou, C.M Androna, A. Zafeiropoulos, SDGDetector Gitlab Repository. (2023). Available at https://gitlab.com/netmode/sdg-detector.
[27] I. Loshchilov, F. Hutter, Fixing weight decay regularization in adam, CoRR abs/1711.05101 (2017). URL: http://arxiv.org/abs/1711.05101. arXiv:1711.05101.
[28] I. Mandilara, E. Fotopoulou, C.M Androna, A. Zafeiropoulos, SustainGraph Gitlab Repository. (2023). Available at https://gitlab.com/netmode/sustaingraph.