Comparative Analysis of Transformer-based Language Models for Text Analysis in the domain of Sustainable Development

NABIL SAFWAT

A THESIS SUBMITTED TO THE FACULTY OF GRADUATE STUDIES IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTERS OF ARTS

MASTER OF ARTS IN INFORMATION SYSTEMS AND TECHNOLOGY

YORK UNIVERSITY

April 2023

Abstract

With advancements of Artificial Intelligence, Natural Language Processing (NLP) has gained a lot of attention because of its potential to facilitate complex human-machine interactions, enhance language-based applications, and automate processing of unstructured texts. The study investigates the transfer learning approach on Transformer-based Language models, abstractive text summarization approach, and their application to the domain of Sustainable Development with the goal to determine SDGs representation in scientific publications using the text summarization technique. To achieve this, the traditional transfer learning framework was expanded so that: (1) the relevance of textual documents to specified text can be evaluated, (2) neural language models, namely BART and T5, were selected, and (3) 8 text similarity measures were investigated to identify the most informative ones. Both the BART and T5 models were fine-tuned on an acquired domain-specific corpus of scientific publications extracted from Scopus Elsevier database. The relevance of recently published works to an SDG was determined by calculating semantic similarity scores between each model generated summary to the SDG's description. The proposed framework made it possible to identify goals that dominated the developed corpus and those that require further attention of the research community.

Acknowledgement

I dedicate this thesis to the memory of both my beloved paternal and maternal grandmothers, Ayesha Khatun and Rokeya Begum, who have passed away during the course of my studies. I will always cherish their memories, and be appreciative of their love and support.

This thesis would not have been possible without the continuous support and motivation from my supervisor, Dr. Marina G Erechtchoukova. I have greatly benefited from Professor Marina's experience and knowledge in the field and have learned a lot under her direction. She consistently goes above and beyond to support my success, which is incredibly motivating. I would further like to thank the committee members Dr. Enamul Hoque Prince and Dr. Andrea Podhorsky for their valuable time towards the thesis defence. Finally, I also want to thank my family for their continuous support throughout the duration.

Table Of Contents

List Of Tables

List of Figures

1 Introduction

As Artificial Intelligence (AI) technology advances, machines are now able to interpret human language with great efficiency. There are multiple sub-areas in AI which are widely explored in research and one such area is Natural Language Processing (NLP). NLP is an interdisciplinary area that deals with automated text processing in a way that imitates human understanding of written texts or spoken sequences of words.

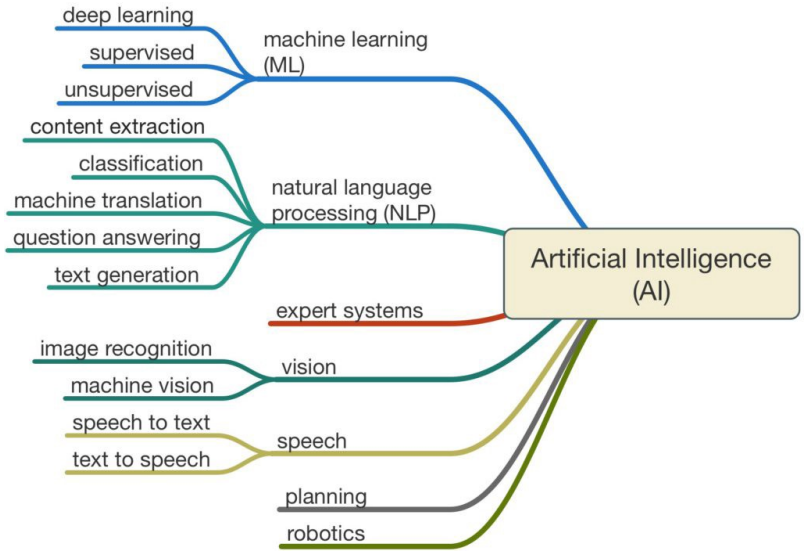


Figure 1. Application areas of Artificial Intelligence (Kruglyak, 2021)

NLP is based on language models. These models utilize various statistical or machine learning techniques to determine probability of occurrence of a given sequence of words in texts. These models after extensive training, gain the ability to predict human language. They can be divided into two categories: Statistical models and Neural Language models. Statistical models are probability based models that help with predicting the next word in the sequence. Latent Dirichlet Allocation (LDA) and Latent Semantic Indexing (LSI) are two commonly used statistical models. Both these models are often used for classification, categorization and

summarization of documents and are widely used as Topic Modelling algorithms. Neural Language Models are developed using Neural Networks (NNs). These models are able to execute complex NLP tasks such as speech recognition, text summarization and machine translation (Taylor, 2021). Previously, Long Short-Term Memory (LSTM) models (Hochreiter & Schmidhuber, 1997) representing a subclass of Recurrent Neural Networks (RNNs) were used to accomplish these tasks. However, in the present day, models based on the Transformer architecture (Vaswani et al., 2017) are leading the trend with state-of-the-art performance. They were previously trained on a large corpus of unsupervised data. This process is known as pre-training the model. The pre-training allowed these models to learn general language structures like how humans would interpret written texts. These models can be applied to various domains using a technique called transfer learning. Transfer learning is the process of using a model previously trained on a set of data on a specific task for another related task (Bozinvski & Fulgosi., 1976).

A transformer is a deep neural network which learns context and meaning by tracking relationships in sequential data. The architecture applies an evolving set of mathematical techniques, called self-attention, to detect subtle ways to link relationships between distant data elements in a series influence (Merrit, 2022). Emerging work suggests a spectrum of new uses for transformer-based language models in AI domains, including time-series forecasting, training machines to discern emotions in speech (Ornes, 2022) and even convert text and speech in real-time. Models based on LSTMs have been popular before the introduction of transformers. The LSTM models rely on a feedback mechanism unlike the transformers which use feed-forward mechanism to train the weights. However, LSTMs rely on sufficient training and testing data from the same distribution. The performance

also varies between the tasks which creates uncertainty and loss of performance. Hence, researchers in the field have realized the limitations of applying transfer learning in RNNs.

NLP is a major application area for transformer-based language models. These models can be applied to a variety of NLP tasks, outperforming RNNs on majority of them. In the present day, several transformer-based language models exist that utilize the full power of the novel encoder-decoder architecture. Among them, models such as BERT (Devlin et al., 2018), GPT2 (Radford et al., 2019), XLNET (Yang et al., 2019) and T5 (Raffel et al., 2019) are commonly used by researchers. The models have several unique characteristics which helps determine their application. Models like GPT, T5 and BART are considered state-of-the-art for generation tasks, whereas BERT and XLNET perform best when used for classification and regression tasks.

Sustainable Development of human society has been declared as the top priority by the United Nations (UN). UN 2030 Agenda had been articulated in 17 interrelated Sustainable Development Goals (SDGs) covering various directions of societal transition towards a better future. However, successful implementation of these goals requires monitoring and evaluation based on large sets of quantitative and qualitative targets and their indicators. Application of NLP techniques for analysis of documents in the area of Sustainability is expected to uncover important information on the attainment of the SDG standards and to become an important tool for sustainability assessment (Matsui et al., 2022). However, methodologies and unified frameworks for AI application to this domain are yet to be developed. In addition, the multi-disciplinary nature of this important domain makes the investigation of language models tested on benchmark datasets an interesting problem.

The domain contains a large amount of unstructured text that require labelling for explicit interpretation. Mapping relevant documents to the SDGs can further aid in faster attainment of the goals. Recent studies have showed that there are existing research within the area that utilize multi-class classification techniques to determine relevancy of documents to the SDGs. However, techniques that utilize specific semantic similarity metrics to determine semantic meaning between given words may be more useful as they may be able to capture much more complex relationships among these words.

The study makes the following contributions:

• Develops a framework for semantic similarity analysis of short texts using transfer learning, abstractive summarization, transformer-based language models, and semantic similarity measures.

• Determines appropriate transformer-based models and semantic similarity metrics for their utilization in the proposed framework.

• Applies the proposed framework to a developed corpus of scientific publications in the area of Sustainable Development.

• Utilizes the proposed framework to ascertain the extent of the relevance of publications to Sustainable Development Goals. This allows to to determine SDGs representations in a corpus of recently publications and identify the current research trend.

2 Literature Review

2.1 Transfer Learning

Transfer learning involves utilizing a pre-trained model that was initially trained on a particular task and applying it to a related task. The process of transfer learning can be divided into two different approaches known as Feature Extraction and Fine-tuning. In Feature Extraction, a pre-trained model is obtained and the final layer weights are updated to generate predictions for the new task. Fine-tuning is another approach to transfer learning where the model is changed to fit the new task and one or more top layers are unfrozen depending on the task requirements. In general, freezing a layer prevents its weights that was obtained from pre-training from being modified. Other layers are kept frozen as the previously learnt knowledge is kept intact and the model does not require to be trained from scratch. Before fine-tuning can be applied to any model, it has to be pre-trained on a large corpus. The study explores pre-trained transformer-based language models and the fine-tuning transfer learning approach. These models have a common goal, to learn deep language structures using their unique methods. Because a pre-trained model had been already trained on a very large corpus of text documents, it requires less resources for fine-tuning and makes it computationally feasible to apply the model successfully to a specific domain's use. The technique can be easily applied on most of the models based on the transformer architecture. These models have their own unique features and training schemes however, the process of transfer learning is similar on majority of the models.

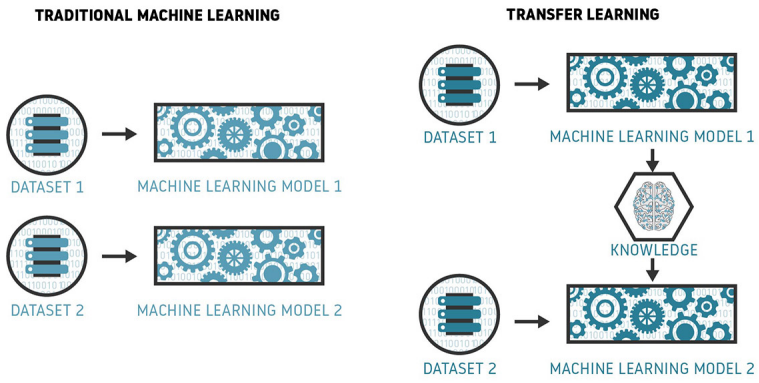


Figure 2. Traditional ML vs Transfer Learning (Martinez, 2021)

Figure 2 above draws a comparison between traditional machine learning approach and the transfer learning approach where two machine learning models share knowledge using transfer learning.

2.2 The Transformer Architecture

The use of Long Short Term Memory (LSTM) in Recurrent Neural Networks (RNNs) had been persistent in the field of text analysis. However, even though these models demonstrated good performance in various applications, they were unable to maintain contextual dependencies for longer sequences as inputs were injected one at a time. That was until (Vaswani et al., 2017) introduced the Transformer architecture which revolutionized the text analysis scene. Nowadays, majority of the models are based on the architecture. These models are faster because the sequences were injected altogether whereas in older models, word-by-word processing was time consuming. The models inherit an encoder-decoder architecture; an encoder reads an input sequence entirely and encodes the text to a fixed-length internal representation. A decoder then used this internal representation to output words until the end of sequence token is reached (Brownlee, 2018).

The architecture consists of multiple multi-head attention layers and feed forward layers. Multi-head Attention is a module for attention mechanisms which made inputs run several times in parallel. A feed forward layer adds non-linearity to the linear transformations of the multi-head attention modules. Lastly, the add and norm layer helps with normalization and deals with the vanishing gradient problem by creating a shortcut between the input and the output of the sub-layer.

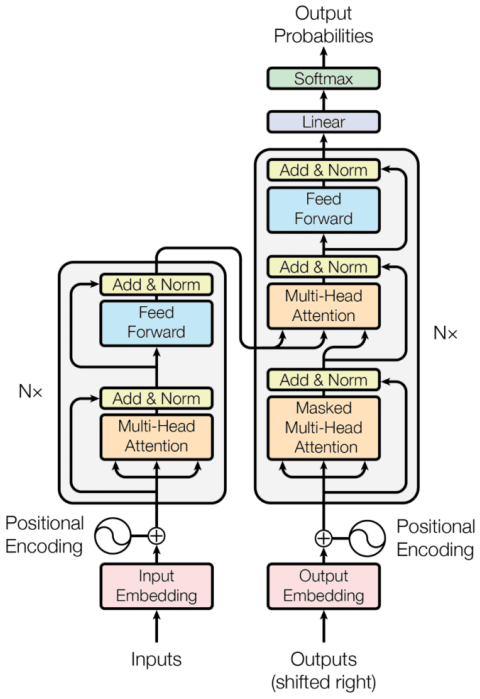


Figure 3. The Transformer Architecture (Vaswani et al., 2017)

An attention function can be described as mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. The authors of the architecture introduced 'self-attention' which is an improved version of the attention mechanism. Previously, the 'attention' mechanism (Bahdanau et al., 2014) was used to address the bottleneck problem that had risen with the use of a fixed length encoding vector, where the decoder would have limited access to the information provided by the input. This was thought to become espe-

cially problematic for long and/or complex sequences, where the dimensionality of their representation would be forced to be the same as for shorter or simpler sequences. However, in the 'self-attention' mechanism, every word is 'aware' of other words in the same sequence and creates a vector representation with respect to other words. This way several vector representations for a single word are attained which are added together to get the weighted average. The output was a single vector much like the attention mechanism but maintains long term contextual dependency.

A scaled dot-product attention utilizes a single attention function using d_{model} dimensional query Q, key K and values V. The attention can be calculated as:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{1}$$

where $\sqrt{d_k}$ is the dimension of the key vector k and query vector q. $Softmax$ classifier converts the values using an activation function and the probabilities are predicted.

The multi-head attention linearly projects the queries, keys and values h times with different, learned linear projections to d_q, d_k and d_v dimensions respectively. Then each of these projected queries, keys and values are processed in parallel, yielding dv dimensional output values.

$$MultiHead(Q, K, V) = Concat(head_1, ..., head_h)W^O \tag{2}$$

where

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \tag{3}$$

where the projections are parameter matrices $W_i^Q \in R^{d_{model} \times d_k}$, $W_i^K \in R^{d_{model} \times d_k}$, $W_i^V \in R^{d_{model} \times d_v}$ and $W^O \in R^{hd_v \times d_{model}}$. R refers to real numbers. Each $head$ is a scaled dot-product from a single function. Finally, the obtained set of $head$ are concatenated and projected once again, resulting in attaining the final values.

Transformers are faster and more scalable than LSTM models to a great extent. They make NLP tasks applicable to large corpora and real-world applications.

2.3 Types of Transformer-based Models

2.3.1 BERT Model

Devlin et al., (2018) from Google introduced one of the most influential language models. The Bi-directional Encoder Representations from Transformers or BERT is one of the top models that harnessed the full power of self-attention that was introduced in the transformer architecture. BERT consists of only the encoder block of the architecture, it does not need decoders because it was mainly built for tasks such as natural language inference, question answering and classification which did not rely on text generation. Therefore, it only needs to encode the language representations. Unlike other transformer-based models, BERT is not auto-regressive i.e., it is bi-directional and understands contexts from both sides simultaneously.

BERT's training consisted of two different sets of tasks. The first was the Masked Language Modelling (MLM) task which hid about 15% of the words in a sequence. The model then tried to predict the hidden (or masked) words based on the available context. In this case, 80% of the time, a word was replaced with a mask, 10% of the time with a random word from the corpus and 10% of the time it was unchanged. This was done to bias the representation to the actual observed word. The second training task was Next Sentence Prediction (NSP). With two sentences A and B, the chance whether B followed A was 50%. Otherwise, it would be a random sentence from the corpus. BERT's inputs has 3 special representations including token, segment, and positional encodings. These are combined to get a vector representation which is fed into the encoder. BERT was pre-trained on very large corpora using the BookCorpus and Wikipedia datasets.

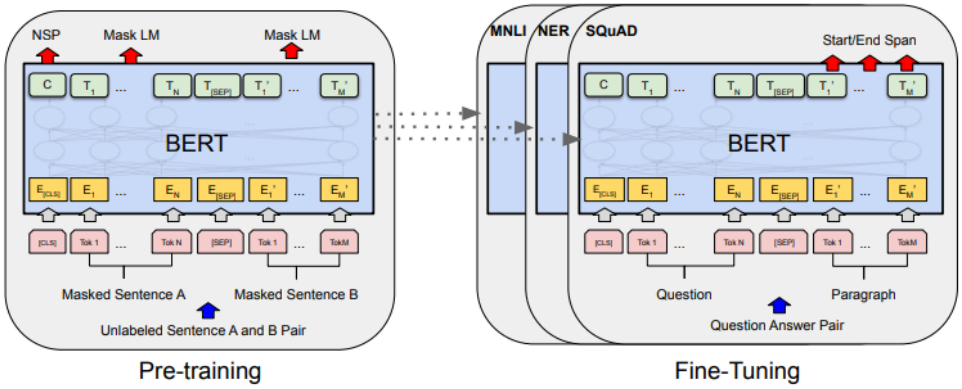


Figure 4. BERT's Pre-training and Fine-tuning process (Devlin et al., 2018)

Apart from the output layers, the same architectures can be used in both pre-training and fine-tuning. During fine-tuning, all parameters can be fine-tuned. CLS is a special symbol added in front of every input example, and SEP is a special separator token for separating parts for text i.e., questions and answers. The number of layers i.e., Transformer blocks are denoted as L, the hidden size as H, and the number of self-attention heads as A. The model is available in two different sizes: $BERT_{BASE}$ (L=12, H=768, A=12, Total Parameters=110M) and $BERT_{LARGE}$ (L=24, H=1024, A=16, Total Parameters=340M).

BERT was tested on tasks such as classification, word predictions and textual analysis where it provided good results and improvement in both short and long text cases. However, one of the major limitations of the BERT model, it is not auto regressive and separates its tokens during prediction. Generating bigrams in this case is not very efficient as when two words are connected to form meaning, they are as independent words by the model. For example, "New" and "York" together form a two-word city but when it comes to prediction in BERT, the model may predict "Los York" because it may learn the words 'Los Angeles' from the corpus. The tokens are also processed separately which causes these words to

lose meaning.

2.3.2 GPT-2 Model

GPT-2 proposed by (Radford et al., 2019) at OpenAI is considered the best text generation model by many. It is based on the older model GPT for generative pre-training. The researchers believed that any language models that was trained with unsupervised learning, did not require any prior understanding of the task to be performed with the model. The models learnt these tasks without any explicit supervision or past knowledge when it was trained on a dataset consisting of millions of web pages. GPT2 was trained on a much bigger dataset than BERT. The model also out scaled its predecessor the GPT model with 10 times more parameters. The data was collected and filtered from Reddit and was known as "WebText" consisting of 8 million web pages.

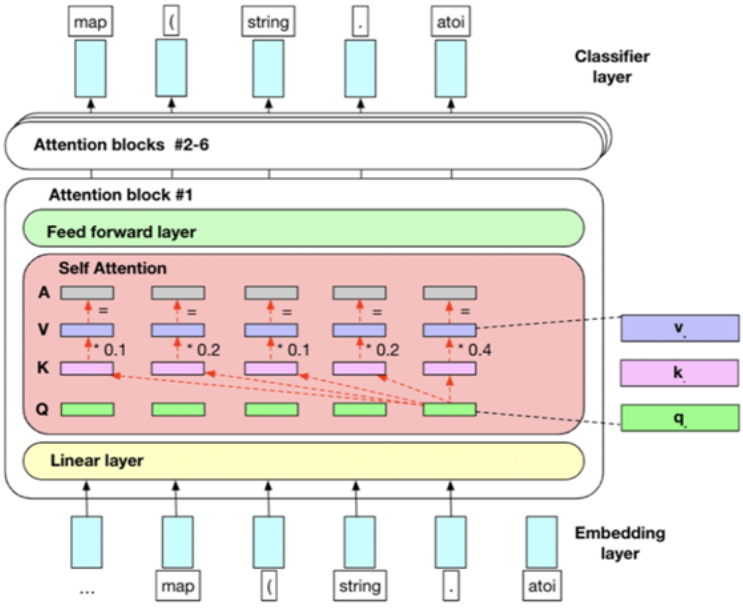


Figure 5. GPT-2 Model Architecture (Radford et al., 2019)

Figure 5 shows the architecture of the GPT2 model. BERT was built stacking

encoder layers on top of each other. However, GPT2 only utilizes the decoder part of the original transformer architecture. Decoders were required as the model excels on natural language generation tasks, unlike BERT which performs better on classification tasks. The decoders acted like encoders and used the multi head self- attention block which dealt with the masking task. The language model had been experimented on several datasets and the zero-shot results had been analysed. It was seen that with the reduction of parameters, the model had a boost in its performance. Overall, the model could perform most of the NLP tasks, but the researchers had talked about making the most of the model by training it on a much larger dataset than WebText. GPT-3 is an extension of the GPT-2 model with much more trainable parameters but is not open-source as of yet.

2.3.3 T5 Model

T5 or Text to Text Transfer Transformer introduced by (Raffel et al., 2019) is another popular model inspired by the Transformer architecture. It is considered the state-of-the-art language model in sequence-to-sequence (seq2seq) tasks. A seq2seq task can be defined as a task whose inputs and outputs are both string of texts. The model is said to be unified as it has the capability to process a variety of NLP tasks simultaneously. BERT could only predict single tokens at once whereas T5 can predict multiple words in a span. Hence, the model was learning to output a span of text sequence rather than a single token.

T5 was trained on the dataset known as “Colossal Cleaned Common Crawl” or C4. C4 was built from scrapping webpages, pre-processing and optimizing the extracted text. A total of 750GB of C4 dataset was used to train the T5 model. The authors generalised the model further with the correct format of data. To achieve

this, they removed markup and non-texts from the data and only used sentences with terminal punctuation. T5 follows similar guidelines to that of $BERT_{LARGE}$ and used d_f = 4096, d_{model} = 1024, d_{kv} = 64 and 16-head attention mechanisms. There exists two variants with $t5_{BASE}$=16 and $t5_{LARGE}$=32 layers each in the encoder and decoder.

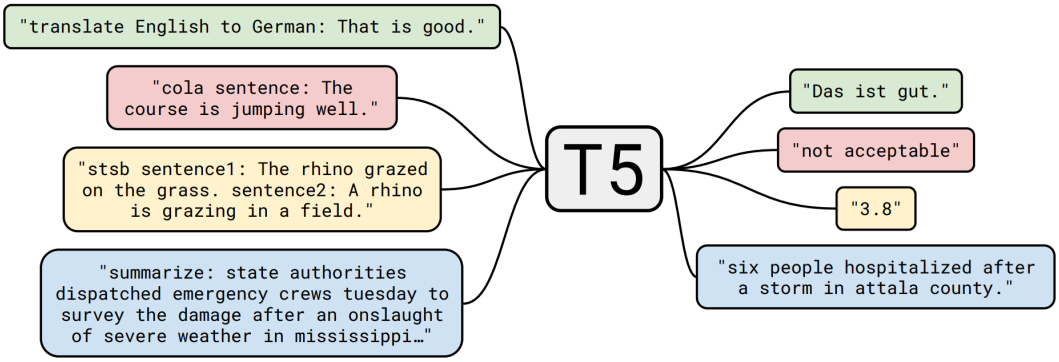


Figure 6. Multi-task approach (Raffel et al., 2019)

Figure 6 highlights that “translate English to German” and “summarise” are two examples of task specific prefixes which allows the model to understand the task and choose the parameters accordingly. Other examples include calculating similarity of two sentences and predicting if a sentence was semantically correct.

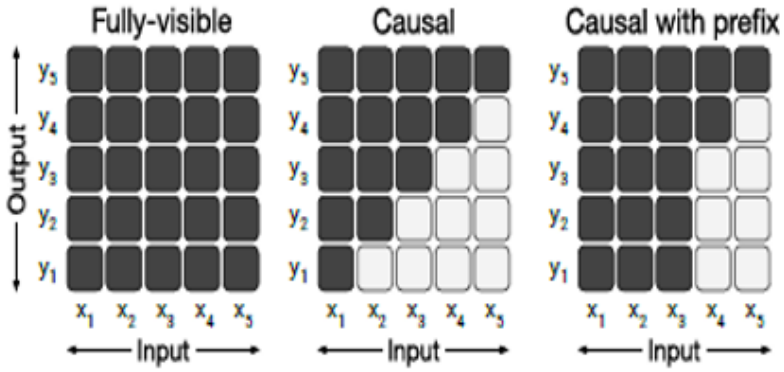


Figure 7. Matrices of 3 different attention masking patterns (Raffel et al., 2019)

In Figure 7, x and y were input and output of the self-attention mechanism respectively. A dark cell at row i and column j indicates that the self-attention mechanism is allowed to attend to input element j at output time-step i. A light cell indicates that the self-attention mechanism is not allowed to attend to the corresponding i and j combination. BERT used the fully visible attention masking pattern where all the tokens would see both the left and right context, which availed bi-directionality, whereas the auto-regressive models used Causal approach where the tokens would only see the past tokens and not the future ones. This method is known as auto-regression. T5 used a causal with prefix approach where the model generates the prefix for the required task, then it behaves like the causal pattern.

T5 hyperparameter selection followed the approach similar to BERT with the exception that it did not mask a single token, it masked a sequence of tokens in the same span. With thorough experimentation, 15% token masking gave the best results. The span lengths were kept smaller because the model had difficulties predicting with less supervision of the surrounding words. T5 provided state of the art results on many NLP tasks with the advantage of being a unified model.

2.3.4 BART Model

BART is another popular language model introduced by (Lewis et al., 2019). It is a denoising auto-encoder which was pre-trained for translation, and other seq2seq tasks like summarization and generation. Like T5, it takes strings as input and output another span of text. The encoder used in BART is similar to BERT's original architecture and is bidirectional. The decoder maintains auto-regression, a left-to-right decoder and the pre-training is unique compared to the other models: it was trained on five tasks: (1) token masking, (2) token deletion, (3) token infilling,

(4) sentence permutation, and (5) document rotation. The model was trained on CNN/Daily mail data for comprehension tasks. It used a noise-added source text as input and used a language model for reconstructing the original text by predicting the true replacement of corrupted tokens. This process was known as "denoising". Noise in a text could be termed as irrelevant or missing data that may not be accurate in the current context. BART also uses 10% more parameters compared to Google's BERT model.

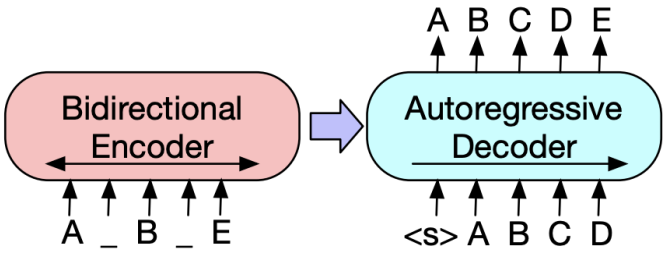


Figure 8. BART Model Architecture (Lewis et al., 2019)

BART utilizes a bi-directional encoder and an auto-regressive decoder. This means that the encoder's attention mask is fully visible, similar to BERT, and the decoder's attention mask is causal, like that of GPT. Some tokens from the text are corrupted randomly and the model tried to denoise the given text. Similar to that of BERT and T5, BART has 2 versions, the $BART_{BASE}$ model which consists of 6 layers and the $BART_{LARGE}$ which has 12 layers of stacked encoders and decoders.

BART showed impressive performance on sequence classification, token classification, text generation and translation and is a go-to model for natural language generation tasks in the present day.

2.3.5 TransformerXL Model

The TransformerXL Model introduced by (Dai et al., 2019) is built on top of the original transformer architecture with some major improvements that avails better results. The original model introduced by (Vaswani et al., 2017) had complexity understanding the full context of the given text. In other words, the model had fixed length context dependency which mostly failed while processing longer texts. In general, longer segments did not respect the boundaries of a sentence, which caused context fragmentation.

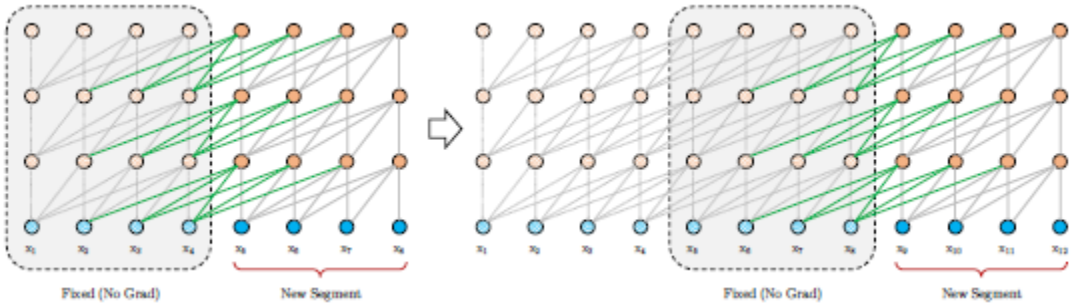


Figure 9. TransformerXL Mechanism (Dai et al., 2019)

The researchers proposed TransformerXL which learns context beyond a fixed length dependency, completely resolving the context fragmentation issue. The XL model was mainly a two-step tune in the training process of the original transformer. The first was a 'segment level recurrence' technique which made sure that previous level segmentation are cached and reused as extended context when the next level representations are being processed. This helped in increasing the largest possible dependency length by an N number of times. In other words, the new segment was 'aware' of the previous segment and these representations were passed forward. This aided in removing the context fragmentation complexity.

$$\tilde{h}_{r+1}^{n} = [SG(h_r^{n-1}) \circ h_{r+1}^{n-1}], (q_{r+1}^{n}, k_{r+1}^{n}, v_{r+1}^{n})] \quad (4)$$

where the function SG(·) stands for stop-gradient, the notation [$h_u \circ h_v$] indicates the concatenation of two hidden sequences along the length dimension. The critical difference between original transformer and transformerXL is that $n+1$ pairs of keys, k_{n+1} and values, v_{n+1} are carried on the extended context h. Here, h_{r+1}^{n} was cached from the previous segment. This mechanism produces a new segment-level recurrency. As h could be kept carrying n times of previous segment, it could carry a lot more than two segments. To achieve the above, a second technique was applied for reuse of the previous states of the segments. It was important to know that the positional encodings were important while reuse so that the previous segments did not lose any context. This was known as utilizing Relative Position Encoding where the actual input to the model was the element-wise addition of the combinations of word embeddings and the positional encodings from (Vaswani et al., 2017).This technique was used to encode the relative positional information of the hidden state. If the previous segment has an input of 0, 1, 2 the new segment would have 0, 1, 2, 0, 1, 2 where the first 3 inputs were from the previous segment and being carried to the next. Even though there were several implementations of TransformerXL on commercial projects, this model still had not been extensively evaluated for application on multi-disciplinary domains.

2.3.6 RoBERTa Model

RoBERTA - A Robustly Optimized BERT pre-training approach by (Liu et al., 2019) was introduced as a modification in the training process of BERT. The researchers realized that BERT was under-trained and with longer pre-training, the model could outperform its previous state-of-the-art results as well as improve predictions on Masked LM task. A combination of different datasets were used in pre-training. There had been changes, tunings in the hyperparameters. While BERT was trained on a 16GB BOOKCORPUS which was a combination of novel books written by unpublished authors and English WIKIPEDIA datasets, RoBERTA was trained on multiple large-scale datasets including CC-NEWS a collection of news articles, OPENWEBTEXT - a combination of web crawled information and STORIES dataset which adds to 100+ GB of data. It was also trained on other smaller-scale datasets.

The implementation of RoBERTA included re-implementation of BERT with similar optimization procedures. BERT had steady learning rate and number of warm up steps at first then it was linearly decayed. In case of RoBERTa, the researchers tuned these hyperparameters separately for each of the settings. BERT was optimised with Adam (Kingman & Ba, 2015) with hyperparameters β_1 = 0.9, β_2 = 0.999, o = 1e-6 and L_2 weight decay of 0.01. The researchers found that tuning β_2 = 0.98 yields more stability and better results. BERT was previously trained on 512 tokens with 256 sequences in each mini batch. It was shown that larger batch sizes improved the overall performance of the model. Bigger batch sizes meant less noise in the gradients and thus the gradient estimate improved. This allows the model to take a better step towards a minimum. However, bigger batch sizes required more memory and each step was more time consuming.

RoBERTA shows improved performance compared to the base BERT model in all the 11 NLP tasks that BERT was previously evaluated on. Bigger batch sizes drastically improved model accuracy on the Natural Language Generation (NLG) evaluation metrics. Within large batch sizes, a standard amount of steps worked best. Longer pre-training on much larger datasets showed better results. Dynamically changing patterns and design choices with addition of more data had shown more improvements in the learning rate. Removing the NSP objective had also shown improvements however, it completely took away one of the main objectives of BERT which was predicting the following sentence.

2.3.7 XLNET Model

Lastly, (Yang et al., 2019) introduced XLNET which is auto-regressive but can accomplish a lot more than the traditional auto-regressive models. The model combines BERT's auto-encoding and the technique used in the TransformerXL model. BERT used bi-directionality which made it lose the power of auto-regression, however XLNET can see bi-directional context but without losing auto-regression. The researchers had applied a new technique to achieve this. When BERT used masking to cover the tokens for prediction, it ignored the dependency of the masked positions, hence it suffers from a discrepancy between pre-training and fine tuning. XLNET learns context from two sides from maximising likelihood over all the permutations. This is achieved by changing the order of the sequence at different cycles. The authors used shuffled permutations, a technique to consider all possible orderings of the words in a sequence and then define a certain ordering on a cycle. This way an order that was previously defined, is not repeated. The model also utilizes memory and the relative positional encodings of the TransformerXL model.

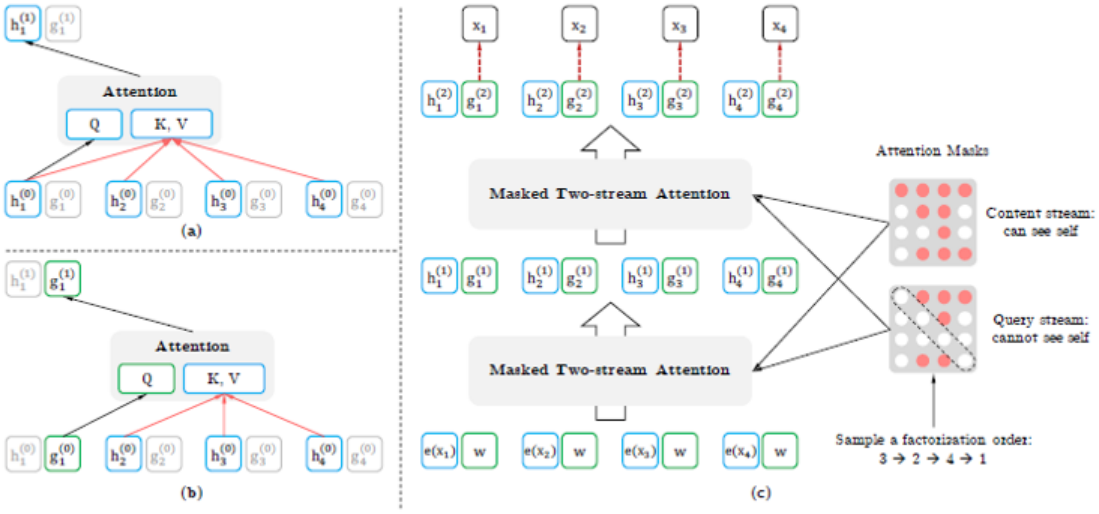


Figure 10. XLNET Mechanism (Yang et al., 2019)

XLNET uses an attention mask to permute the factorization order. As seen in the diagram, to predict content representation of h_1, the model needs to have all 4 of the token information. This was achieved by changing the order of the words in the sequence. The input sequence has one order but the attention mask allows implementation of different factorization orders of the same input sequence. Given an original order h_1, h_2, h_3 and h_4, a random factorization order h_3, h_2, h_1, h_4 in one cycle is achieved. It would have a completely different order in the following cycle. Even though this process somewhat looked like BERT's bi-directionality, it is done with a completely different mechanism to keep auto-regression intact. The unique training approach of XLNET allows it to act as both an auto-encoder and an auto-regressive model.

BERT has a major limitation when processing words which are connected to produce a different meaning. An auto-encoding model would use separate tokens for processing. Hence, it is not always efficient for partial prediction.

$$\mathcal{J}_{\text{BERT}} = \log p(\text{New} \mid \text{is a city}) + \log p(\text{York} \mid \text{is a city}),$$
$$\mathcal{J}_{\text{XLNet}} = \log p(\text{New} \mid \text{is a city}) + \log p(\text{York} \mid \text{New}, \text{is a city}).$$

Figure 11. Partial prediction of BERT and XLNET (Yang et al., 2019)

Considering the two words “New” and “York” which formed the city “New York” would completely lose meaning in BERT as the model separates and predicts these tokens independent of each other. However, XLNET can see the dependency via different orderings. These are some of the major advantages of auto-regressive models over auto-encoding models.

XLNET uses the same hyperparameters of the $BERT_{BASE}$ model in the comparison and shows proficient results on Stanford Question Answering Dataset (SQuAD) and General Language Understanding Evaluation (GLUE) datasets. XLNET overcame BERT with better performance on several NLP tasks including Question Answering (Q&A), Natural Language Inference (NLI) and Information Retrieval.

2.4 Application of Transformers

Transformers revolutionized NLP by significantly improving performance on a wide range of tasks, particularly those requiring longer dependencies. Since its debut, the Transformers have been widely used and expanded in NLP research, resulting in improvements in fields such as language modelling.

The BERT model was applied to short answer grading task by (Sung et al., 2019). According to the authors' hypothesis, the contextual representations of pre-training is the key to efficient predictions. For fine-tuning on grading and language tasks, BERT was trained on an English language corpus. Even though this provided some accurate predictions, it would be possible to improve the model training with the resources gathered from textbooks.

The approach consisted of three steps: (1) selecting the domain - grading academic papers; (2) enhancing the pre-training with domain specific knowledge and (3) fine-tuning for short answer grading. Collecting the relevant textbooks for the task was vital in this case. The idea was that the question and answer would be in the same paragraph. This was validated with several question answer pairs and manually examined to check. 90% of these pairs had the answer and question in the same paragraph. Next, these pairs were used as input to the BERT model. Incorrect, incomplete, and irregular answers were ignored as they might have damaged the learning phase of the model. A total of three question answer datasets were used in a multi-class classification task where it would predict whether the answer is correct, partially correct or incorrect. Pre-training with domain textbook information positively affects the performance of a model in a domain. Fine-tuned BERT with the textbook data provided much better results on the grading task.

(Harly et al., 2021) explored quantitative argument summarization of textual data. This research involved collecting arguments and providing summarizations of these arguments with highlighted key points. The authors mapped the important arguments to a key point which can be termed as a high-level argument. The implementation was like previous works on quantitative argument summarization with major tweaks in pre-processing steps and upgrades on hyperparameters during the training phase. The authors used IBM ArgQ, a large-scale benchmark dataset for the task of mapping arguments to key points. T5 and RoBERTa were base models for the task; the number of hyperparameters did not exceed significantly in any of these models which was then fine-tuned on a large number of training steps with a higher batch size. The evaluation included calculating the matching score between the argument and the key point from the original model output y. It was seen that mapping the arguments to key points showed higher performance in quantitative summarization on the T5 model. The authors further concluded that increase in number of model hyperparameters would show even better performance.

(Laskar et al., 2020) explored contextualised embedding-based transformers for sentence similarity modelling. The task was to predict the most appropriate answer among a few selections for some given questions. Techniques for sentence similarity were explored in the past, however those technique made the words lose their context in multiple sequences. Most of the transformer-based models that are similar to BERT use contextualised word representation which deal with context fragmentation in a sequence. The authors integrated the contextualised word embeddings of the transformer encoders with the feature based and fine-tuning approaches.

For the feature-based approach, the positional embeddings and the contextualised embedding representations were combined to maintain the sequence's order. The encoder calculated the values and passed them through the feed forward and pooling layers to attain the condensed vectors. The decoder returned words from these vectors and calculated the cosine similarity between the sentences. Both the BERT and the RoBERTa models were used in the process with some hyperparameter upgrades. There were six different datasets used for the application of the approaches. Two of them were specifically question answering datasets and four community question answering datasets scrapped from social media websites.

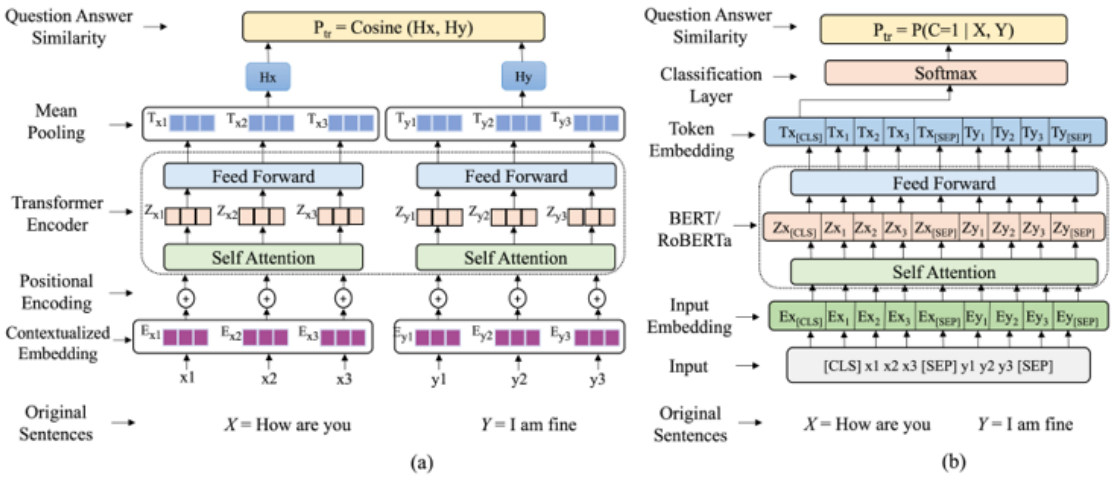


Figure 12. Proposed (a) Feature based and (b) Fine-tuning Approaches (Laskar et al., 2020)

The training of the BERT and RoBERTa models had similar settings to that described in the original papers. Both size variations of these models were fine-tuned for the pairwise sentence classification task. The implementation was based on PyTorch, a python-based machine learning framework. There was another implementation on ELMo model, an LSTM based language model and it was

seen that both BERT and ELMo embeddings improved the performance of the feature-based approach on all of the six datasets. In the fine-tuning approach, the authors also implemented the XLNET language model and in comparison with BERT, it was seen that XLNET outperformed BERT on the question answering datasets. $BERT_{Large}$ outperforms $XLNET_{Large}$ on the YahooCQA and SemEVAL 2016 datasets but the opposite was seen in the older versions of the datasets. $RoBERTa_{LARGE}$ model had the best performance of all the models and achieved state of the art results. In terms of comparison between the fine-tuning and feature extraction approaches, fine-tuning showed significantly better results than the feature-based approach.

(Chen et al., 2021) detected fake news information using CT-BERT, a domain specific model for fake news detection. Because of short text sparsity and lack of semantic meaning, traditional models did not perform well on the constructed dataset. The approach required expansion of the token vocabulary to find the semantics of the text provided, followed by adapting the $Softmax$ loss to find common samples for fake news and lastly one multi-layer perceptron to integrate the high-level representations. The predicted features were extracted using RoBERTA and BERT.

Figure 13. Modules attached to each stage of the fine-tuning (Chen et al., 2021)

The model was mainly detecting if a given news is true or false. Given a sentence x = $t_1, t_2, t_3, t_4..t_n$ the predictor had to guess a correct label y. The proposed network was derived from the original BERT model with integration of several modules that enhance the Natural Language Inference (NLI) task performance. These modules involve three sets of approaches. (1) Training with additional tokens. These were the six most widely used words for Covid-19 (covid-19, covid19, coronavirus, pandemic, lockdown and virus) were counted in training and added to the existing vocabulary of the CT-BERT model. (2) The model was optimized using the "heated up" SoftMax loss function. The heating up of the classifier can be termed as training the model based on classification error calculated by cross-entropy between

$Softmax$ layer output and one-hot vector of ground truth. Training these classifiers with different temperatures of the $SoftMax$ function led to different distributions of the embedding space (Zhang et al., 2022), and (3) a gradient-based adversary training was implemented before being fed to the CT-BERT and RoBERTa vocabulary.

The works of (Liu et al., 2019) explored zero-shot text summarization where they introduced SummAE, a denoising auto-encoder which was trained using an unsupervised approached. The results showed that traditional auto-encoders do not perform well on summary generation and can be outperformed by addition of a novel self-supervised pre-training and introduction of new denoising schemes.

(Arslan et al., 2021) conducted a comparison of the pre-trained language models in multi-class short text classification. In this study, the authors considered 'FinBERT' by (Araci, 2019) as the benchmark model as it was originally proposed to analyze sentiment of text sequences in the financial area. The pre-trained models including XLNET, BERT, DistilBERT, RoBERTa and XLM were compared with the benchmark. The models from the HuggingFace Transformer library (Wolf et al., 2019) were fine-tuned for the classification task. Similar batch sizes were used for both training and testing, sequence length of 128 and with a low learning rate. There were two large-scale and two smaller datasets used in the fine-tuning phase.

The authors concluded that FinBERT model, even with adapted vocabulary did not show any improvements in comparison to the BERT-based models. It was seen that the RoBERTa model achieved the best scores on the majority of the datasets for each metrics whereas other models demonstrated lower performance. The performance for majority of the models were not up to the mark in the two large-

scale datasets while achieving better results on the smaller datasets. The performance of the base pre-trained language models were high except on the big datasets where there is a decrease in performance. As a model with the best results and accuracy, RoBERTa was used for comparison with the FinBERT model. The authors expected that RoBERTa would show better results than FinBERT on the non-financial datasets and it did. However, RoBERTa also showed similar performance to FinBERT on the financial datasets. This lead the authors to the conclusion that the model introduced for the financial domain does not outperform robust pre-trained models such as RoBERTa. FinBERT used an adapted vocabulary on the pre-training of multi-class text classification which according to the original author of FinBERT provided better performance in the classification. However, researchers had determined that the adapted vocabulary on pre-trained FinBERT model did not show better performance than what could be obtained with BERT's vocabulary for the classification task of financial documents.

Transformers have also been investigated in tasks outside the purview of NLP, such as time series forecasting, and classification. Traffic flow forecasting had been a tricky task for the traditional models. LSTMs do a good job at time-series tasks which deals with making predictions based on data available at a certain time and had been used to do most traffic-based predictions. (Reza et al., 2022) introduced a five head, five identical layers of encoder and decoder transformer-based model to achieve accurate traffic flow forecasting predictions. The model used a multi-head attention which allowed the query Q, key K and value V representations to be projected linearly and multiple times in parallel similar to the base models but with increased multi-head channels. The model further included the Square Subsequent Masking technique which was used to prevent the atten-

tion mechanism from showing some of the exact values to the decoder. Decoders should only be able to attend to its past values and predict the future values.

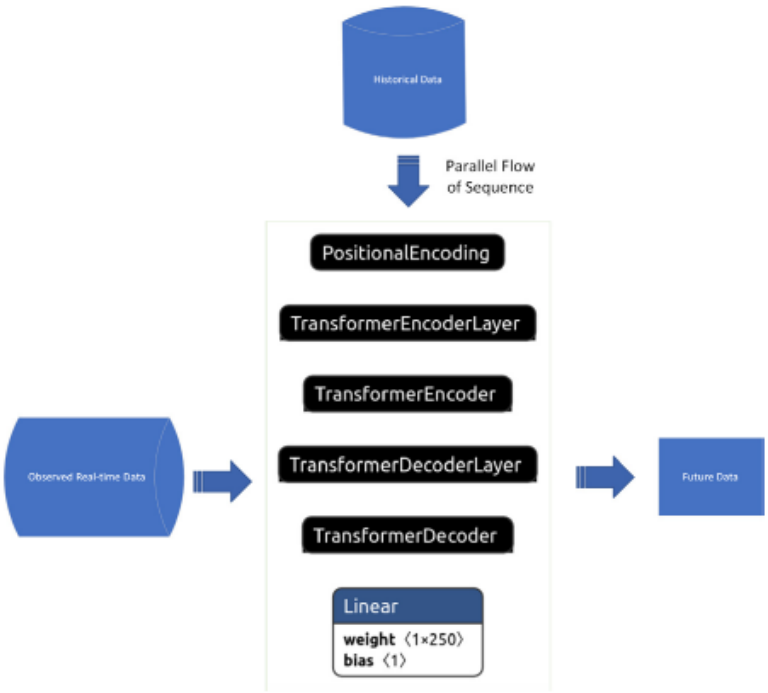


Figure 14. Proposed Model (Reza et al., 2022)

Historical and Real-time data were combined to form the corpus where the historical data was only fed to the encoder for training. The decoder was being fed the observed real-time data which together would predict the future of traffic forecasting. The model had a higher success rate over the LSTM and RNN models.

(Brinkmann & Bizer, 2021) explored hierarchical product classification using domain specific language modelling. The task was to collect product offers from different product aggregators like marketplaces and integrate them into a single hierarchy of product offers. The authors used the RoBERTa model to show that the model performs better than the traditional FastText based classification technique by collecting product offers and categorising them. The second task was improv-

ing the performance of the model which included self-supervised pre-training from a corpora of product offers. The model utilised the 'Common Crawl' corpus similar to that of T5, which was a mixture of scrapped product titles, their description along with the categorical hierarchy of the website that had the product.

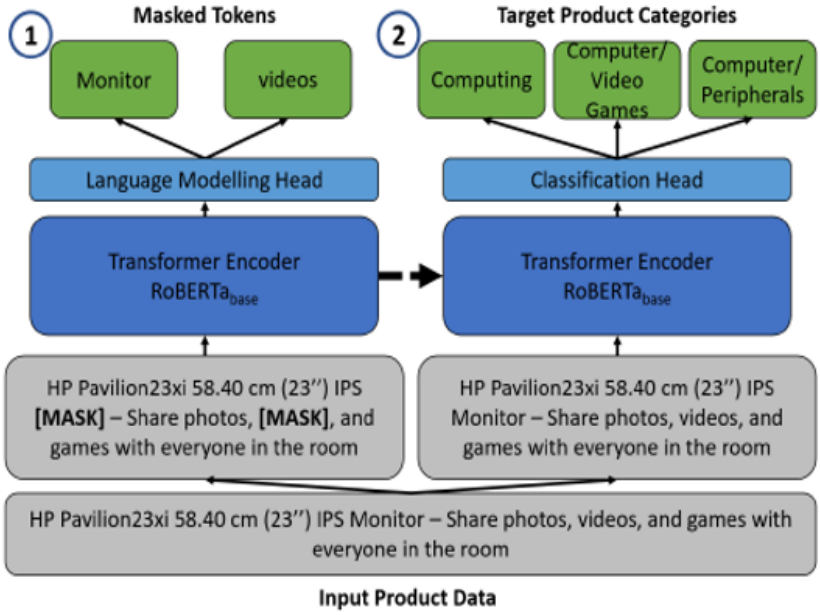


Figure 15. Overview of (1) Pre-training and (2) Fine-tuning phase (Brinkmann & Bizer, 2021)

Thorough experimentation on the base RoBERTa model showed promising results. The model showed good performance when evaluated along with other RNN based models.

The encoder-decoder architecture had been the subject of numerous studies on classification tasks. (Liu et al., 2021) proposed EncT5, a model to efficiently fine-tune a pre-trained T5 model for classification and regression based tasks by using the encoder layers. The decoder was completely removed and two new components were introduced in its place - a pooling layer and a projection layer. A pooling layer is used to down-sample the spatial dimensions i.e., height and width

of the input feature maps. It lowers the model's computational expense and aids in enhancing the input data. By discarding less pertinent data, a projection layer develops a low-dimensional representation of the input that captures the most crucial properties. In summary, the input data is projected onto a lower-dimensional subspace by the projection layer. Both T5 and EncT5 were pre-trained on similar data and experimentation was done on the T5 1.1 Checkpoint available to download on the 'Huggingface' platform. This checkpoint was pre-trained without mixing NLP tasks which showed better generalisation of the model. The authors hypothesised that decoder weights from the first layer and the target embedding weights after the first layer loaded from the decoder were not compatible with each other. Hence, the decoder was removed and the model randomly initialised the class projection in the projection layer. In terms of hyperparameters, a large batch size was used and the best checkpoint was collected for each task. Best checkpoint can be termed as the best possible weights that can be achieved by a model after a training epoch. EncT5 showed superior performance over T5, with reduced number of hyperparameters.

2.5 Related Work in the Sustainability Area

To fulfill UN's sustainability plans to reach sustainable society and environment within 2030, multi-disciplinary efforts are required to transform the development process. NLP could act as a powerful tool that aids in this transformation process. However, various ways for NLP applications within sustainable development are yet to be investigated. (Matsui et al., 2022) and (Guisiano et al., 2022) use multiple data sources for mapping texts to SDGs. However, these applications lack persistent data collection methods and techniques that could allow scalability.

(Matsui et al., 2022) presented an NLP model which utilized the BERT model. The authors argued that NLP models could be used to overcome the challenges of elaborating and visualizing the complex inter-connections between the SDGs as well as connecting stakeholders to facilitate collaborative action. The authors reviewed the literature on the use of NLP in the context of sustainable development. However, it was realized that even though there were some existing research of the domain in tasks such as sentiment analysis, stakeholder analysis and knowledge representation, there was a lack of research on the use of NLP to support SDGs specifically. The model consisted of three main components - (1) A semantic translation module, (2) a nexus visualization module and (3) a stakeholder connection module. The semantic translation module elaborated the SDGs into interpretable language for a wider audience, while the nexus visualization module visualized the interconnections between the SDGs, grouping them to determine their similarity. The stakeholder connection module connected stakeholders.

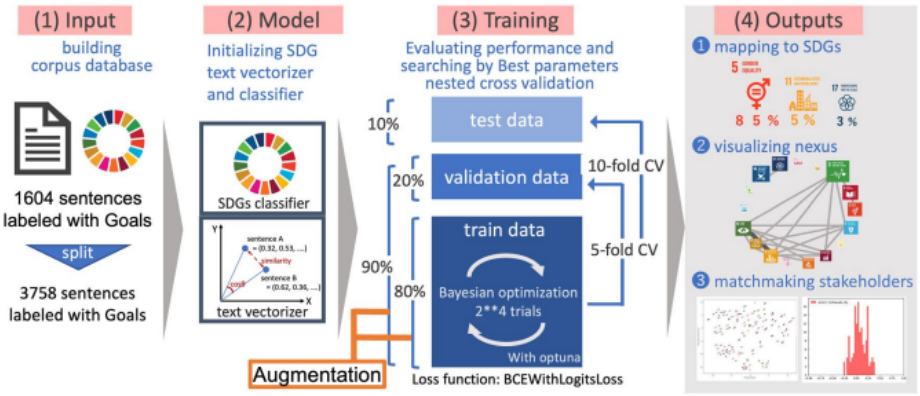


Figure 16. 4 step structure of the model (Matsui et al., 2022)

Corpus was built using documents published by official organizations and multi-labels corresponding to SDGs. A pre-trained Japanese BERT model was fine-tuned on a multi-label text classification task, while nested cross-validation was conducted to optimize the hyperparameters and estimate cross-validation accuracy. A system was then developed to visualize the co-occurrence of SDGs and to couple the stakeholders by evaluating embedded vectors of local challenges and solutions.

The authors tested the NLP model using a case study in Japan. They found that the model was able to effectively accomplish of the three tasks listed in Step 4. They also found out that the model had the potential to support decision-making and policy formulation relating to the SDGs.

Another study conducted by (Guisiano et al., 2022) focused on tracking and reporting of the progress towards achieving the SDGs and suggested categorization of large amounts of textual data, such as news articles, research papers, and social media posts, according to their relevance to SDGs goals. This task is often time-consuming and requires expert knowledge, making it difficult to scale. To address these challenges, several studies had proposed the use of deep learn-

ing techniques for automatic classification of texts from the Sustainability domain. One such tool is the SDG-Meter, which is a deep learning based tool for automatic text multi-class classification of the SDGs.

Figure 17. SDG-Meter interface (Guisiano et al., 2022)

One advantage of the SDG-Meter is that it did not require expert knowledge or manual annotation of the text data, which made it more scalable and efficient compared to traditional methods of text classification. Additionally, the tool could be easily adapted to other domains and languages, making it a versatile solution for text classification tasks. Overall, it was said to be a promising tool for automatic text classification of the SDGs, and could be a valuable resource for tracking and reporting on progress towards achieving the goals. Further research would be required to evaluate the performance of the tool on a wider range of text data and languages, and to investigate its potential applications to other domains.

(Vinuesa et al., 2020) discussed the potential role of AI in achieving sustainability.

AI could contribute to SDGs in multiple ways such as improving decision making and addressing global challenges. The goal was to answer the question “Is there published evidence of AI acting as an enabler or an inhibitor for this particular target?” for each of the 169 targets within the 17 SDGs. A consensus-based expert elicitation process was conducted and informed by previous studies on mapping SDGs inter-linkages.

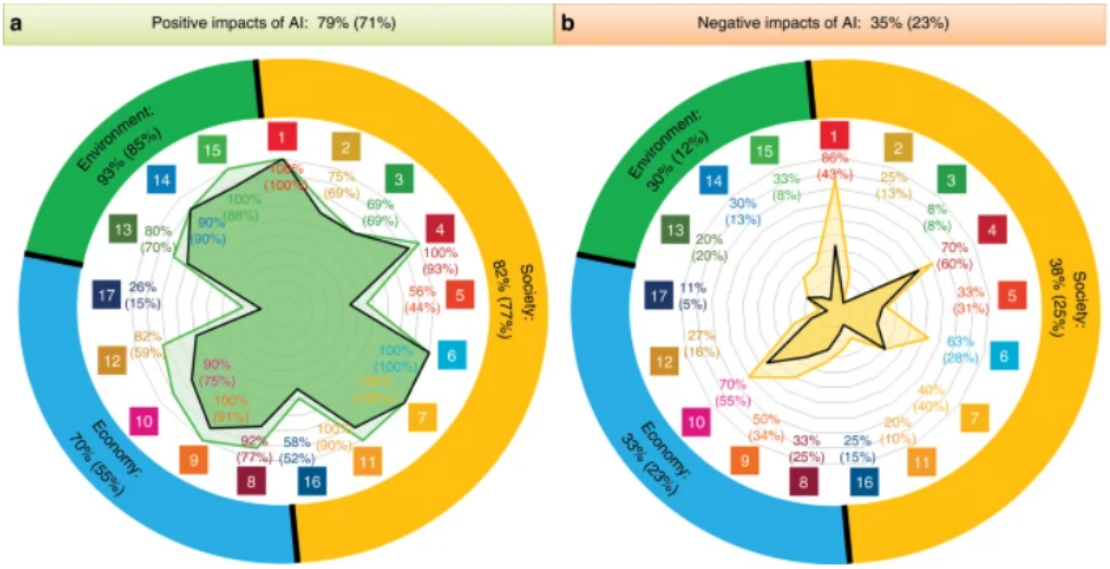


Figure 18. Inter-connection between AI and SDG (Vinuesa et al., 2020)

Researchers claimed that AI based technologies are an enabler for majority of the targets by supporting the provision of food, health, water, and energy services to the population. However, current research priorities have neglected crucial elements. In order to enable sustainable development, the quick development of these applications must be supported by the necessary regulatory insights. Ethical standards could suffer if these are not explored.

NLP applications could also be used for efficient waste management, where most efficient policies concerning e-wastes could be identified and explored. Transformer-

based models were robust hence, they could be fine-tuned in many different unexplored techniques which could be beneficial to both economy and society. Many researchers had explored e-waste and one such study conducted by (Ali, S., & Shirazi, F., 2022) utilize a transformer-based machine learning approach for e-waste management. The study aimed to identify both challenges and opportunities in Canadian waste system and the lessons Canada could adopt from the Swiss management system which the researchers argued was far superior.

A total of 463 research files were downloaded from Scopus, of which 74.3% were full research articles published since 2012. The original BERT model was trained and used as the base model for text analysis using a combination of two datasets. The task was to retrieve the appropriate documents and quantify the relevance of the extracted keywords. Hence, Mean Average Precision (MAP), a popular retrieval metric was used in this case.

AI based research on SDGs and further developments depended heavily on the availability and accessibility of related real-world data collected by the community. However, the sets of data did not include any structure or inter-connections (Spezzati et al., 2022). There was 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. These studies examined various machine learning approaches optimized for NLP-based tasks for classification of existing domain-related reports according to their relevance to the SDGs. (Angin et al., 2022) demonstrated that fine-tuned RoBERTa achieved very high performance in the attempted task, which was promising for automated processing of large collections of sustainability reports for detection of relevance to SDGs.

Existing studies proved that sustainability could benefit from NLP applications. However, existing techniques are not always well-equipped for application in multi-disciplinary domains where the results can be improved. Unified frameworks for application of NLP approaches can be a solution to such problems and can aid to overcome complexities of quantitative evaluation of attainment of on SDGs. However, it is difficult to implement such frameworks.

2.6 Semantic Similarity Measures

The accuracy of model generated outputs can be measured by comparing the generated texts with the existing human written texts. Several approaches have been proposed, however not all metrics are applicable for a particular NLP task. BLEU Score (Papineni et al., 2002) and ROUGE Score (Lin, C., 2004) are two commonly used metrics for natural language generation tasks.

BLEU score measures precision. Precision is a metric mainly used for classification algorithms, but in the NLP context it refers to the fraction of words in the generated translation that are also present in the reference translations. BLEU score assigns a score between 0 and 1 to the generated text, with a score of 1 indicating a perfect match with the reference. A higher BLEU score indicates a higher degree of precision in the generated text, meaning that more of the words in the generated text are also present in the reference. It counts the matching of n-grams to ensure it takes the occurrence of words in the reference text into account. ROUGE Score much like BLEU score assigns a score between 0-1 and measures recall, which is another common metric classification. However, in this case it counts the number of overlapping n-grams i.e., sequences of n consecutive words between the generated summary and the reference summaries. Both these

metrics can be calculated for different n values, such as unigrams (n=1), bigrams (n=2), trigrams (n=3), etc. A higher value of n results in a more strict evaluation, as longer sequences of words must match between the generated and reference text.

These scores are considered appropriate in NLG tasks; however, they do not consider word synonyms, hyponyms, and meronyms. Word relatedness is also ignored which denies a reasonable score to potentially good text generation. Pre-trained models are abstract, and they do not always predict same words. Therefore, methods that consider such words must also be explored.

(Pedersen et al., 2004) proposed measuring semantic similarity using WordNet by considering word synonyms, hyponyms, meronyms and relatedness. This method is based on the lexical database 'WordNet' which uses an ontological structure for creating relationships between words. It calculates similarity based on the path length between two concepts. The shorter the distance between two words in the lexical database, the closer is their meaning (Wu & Palmer similarity). The similarity can be applied to verb and noun pairs. Concepts must be in the same physical hierarchy for a measurement. Sematch by (Zhu, G., Iglesias, C.A., 2017) is a python framework for development, evaluation, and measuring similarity between concepts from knowledge graphs. It allows implementation of knowledge based semantic similarity from structural knowledge in a taxonomy. There are several methods of calculating similarity between words in a taxonomy. These include comparing information content (Resnik, P., 1995), length of the path between words (Wu and Palmer, 1994), or by comparing information commonality and differences (Lin, C., 1998).

Word embedding techniques are crucial in enhancing encoder performance in

deep learning models. Most of these embeddings rely on cosine similarity between concepts in a vector space. The same mechanism can be applied to calculate semantic similarity between words and concepts. Each of the transformer-based models have their own unique input encodings which are crucial to finding words that have the same meaning or are related to each other. ”SentenceTransformer” is an open-source python package which utilizes pre-trained transformer-based models. Models such as T5 (Raffel et al., 2019) and RoBERTa (Liu et al., 2019) are relevant and can be fine-tuned to calculate semantic similarity. Here, inputs are encoded to fixed length representations and their vectors are compared to find word similarity and relatedness.

Two common word embedding based models are Word2Vec (Mikolov et al., 2013) and FastText (Bojanowski et al., 2016). FastText is an extension of Word2Vec and uses n-grams of word embeddings. Both these models convert words to a vector and these vectors are generated in such a way that two semantically similar words are close to each other in the vector space. To find the words that are semantically close to each other, the existing word corpus in Word2Vec and FastText can be scanned to find the corresponding vectors. The closeness can be calculated using cosine similarity and then mapped back into words.

3 Problem Domain: Sustainable Development

Sustainable Development Goals (SDGs) provide directions to combat the urgent environmental, political, and economic needs. Introduced in 2015 at the UN conference at Rio de Janeiro, SDGs explicitly call on all stakeholders to apply their creativity and innovation to address sustainable development challenges. There are 17 goals divided among three categories: environmental, social, and economic. The goals address some of the most pressing global challenges, such as poverty, inequality, anthropogenic impact, and environmental degradation, and aim to create a more sustainable and equitable world. They adopt a holistic approach to recognize that economic, social, and environmental sustainability are interconnected and interdependent. SDGs aim to leave no one behind, regardless of gender, race, ethnicity, or socio-economic status, and ensure that everyone has access to opportunities and resources. The goals also provide a framework for multilateral cooperation and actions, promoting collaboration between governments, businesses, civil society, and individuals to achieve common goals. Furthermore, goal-specific, and measurable targets and indicators are identified, enabling monitoring of the progress toward the attainment of these goals at the global level. SDGs represent a shared vision for a better future and therefore, explore a critical domain of knowledge that must be explored and expanded.

The environmental challenges society faces today are well documented using data of various types from observations to modelling and description in natural languages. However, there exists lack of applications that can make proper use of these data. Whether it's abnormal natural phenomena, over-consumption, inefficient production, or anthropogenic impacts, steps must be taken to safeguard the present and future of the planet through technological innovations that sup-

port increased production, reduced human environmental impact, and informative sustainable decisions and policies to make positive societal changes.

The role of AI in this process can hardly be overestimated due to its ability to process large volumes of data in different formats efficiently, learn from the documented experience, and adjust to new inputs outperforming humans. The UN SDGs are interconnected forming a foundation for a positive change. SDGs are designed to support governments and corporations to collaborate in delivering a common agenda (Prytz, 2022). The current format of proposed SDGs and their targets correspond to the global level. To make them an operational policy framework for transition to a better future, the targets must be translated into indicators at regional and local levels. Without thorough expert and scientific recommendations on the operationalization, the targets may be ambiguous (Hak et al., 2015). NLP applications can aid to sustainability assessment of an undertaking or a policy due to their ability to reduce the cost and time barriers of structuring textual and qualitative data. However, there had been too few NLP applications in SDGs area. (Conforti et al., 2020).

AI technologies in general, offer three main benefits. First, AI permits the automation of important, but repetitive and time-consuming tasks, allowing humans to focus on higher-value work. Second, AI and specifically, NLP techniques reveal insights that are otherwise trapped in massive amounts of unstructured data that once required human management and analysis, such as data generated by videos, photos, written reports, business documents, social media posts, or e-mail messages. Third, AI can interpret data resources to solve the most complex problems. Consequently, NLP capabilities must be used to aid in achieving SDGs.

The lack of NLP applications in the Sustainability area requires identification of

research gaps. NLG tasks condense large amounts of information into a brief, digestible format. Scientific publications describe various techniques and outline major key points which are often ignored by researchers due to time constraints. Hence, applying NLP tasks to the area can help to address some primary issues in SDGs research related to qualitative and quantitative assessment of SDGs attainment, along with identifying these gaps.

4 Methodology

Among the two different techniques embedded in transfer learning, the fine-tuning method was used. A standard fine-tuning framework can be divided into nine stages. These are - (1) Define the Problem, (2) Select Appropriate Task, (3) Select Appropriate Model, (4) Acquire Corpus, (5) Pre-process the acquired Corpus, (6) Obtain Pre-trained Model, (7) Fine-Tune the Model, (8) Generate Predictions on unseen data, and (9) Evaluate Predictions to measure Accuracy. The detailed processes of a standard fine-tuning framework are shown in Figure 19.

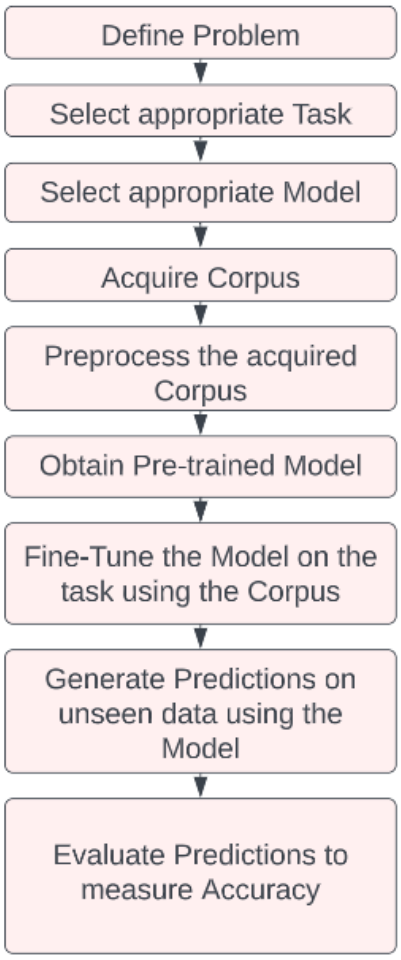


Figure 19. Standard Fine-Tuning Framework

This framework can be utilized to build and evaluate models for multiple tasks such as classification, regression and time-series predictions. However, the standard framework is not applicable for complex NLG tasks such as text generation involving transformer-based models because the necessary data formats entail text documents. Therefore, an NLG task specific framework that fine-tunes models and evaluates generated text is required. This framework is primarily directed towards determination of degree of document relevance to targeted texts in any field. Figure 20 shows the expanded framework for use in this regard.

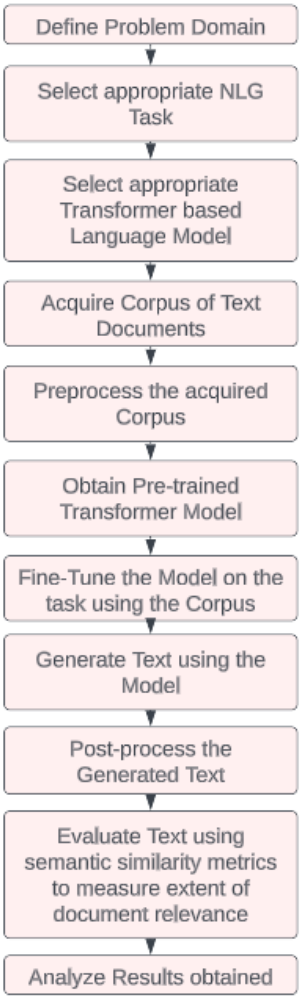


Figure 20. Expanded Fine-Tuning Framework for NLG Tasks

4.1 Defining the Problem Domain

Selecting an appropriate domain for analysis is the initial phase of the framework. Application of transformer-based models to a multi-disciplinary domain and evaluation of their performance is an interesting research task. Application of language modelling within such a domain would be useful and further aid in identifying research gaps and domain-specific knowledge.

4.2 Selecting Appropriate NLG Task

NLG tasks involve generating words or sequences from machine-readable data. These tasks are a major subset of natural language processing (NLP) and are used in a wide range of applications, including machine learning and data analytic. Some popular NLG tasks are text summarization, question-answering, dialogue generation etc. Appropriate task selection is critical for accurate interpretation and validation of the obtained results.

4.3 Selecting Appropriate Transformer-based Model

Not all models utilize the complete transformer architecture. Encoder only models such as BERT (Devlin et al., 2018) and RoBERTa (Liu et al., 2019) are good enough to generate text, but do not have decoders. Models with both encoder and decoder components can overcome the exposure bias problem in NLP since these models generate the entire output sequence at once rather than one token at a time, which is the case in encoder-only models. These include GPT (Radford et al., 2019), BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation (Lewis et al., 2019), and T5: Text-to-Text Transformer (Raffel et

al., 2019). The above mentioned models are widely used for NLG tasks. TransformerXL (Dai et al., 2019), an extension of the original transformer architecture and XLNET (Yang et al., 2019) have claimed to achieve state-of-the-art results on several NLP benchmarks, but their architectures have not been extensively evaluated with applications in the field.

4.4 Acquire Corpus of Text Documents

The acquired corpus must provide an accurate representation of the topics and techniques associated with research in the domain. Steps include selection of data sources that cover a wide range of research publications in the selected domain, and identifying appropriate keywords to search for relevant scientific publications on a given time frame. If the corpus does not represent the recent topics and techniques covered in the selected domain, this may directly affect model performance on text generation. Choosing a time period when compiling pertinent articles would offer intriguing insights on subjects covered at that time.

4.5 Pre-processing the Corpus

Text documents require transformation before they can be used as input data to the model. Pre-processing is done based on the requirements of the task at hand. Some of these steps include combining textual data into a single file, removing inappropriate punctuation and illegal characters, and renaming column names. Lastly before commence training, the filtered data needs to be separated into train, test, and prediction set. Certain encoder-decoder models require a ”prefix” column, which informs the models about the task at hand. Some examples of prefix include ’translate English to German’, ’summarize’, ’classify’ etc., which point

towards a particular NLP task.

4.6 Obtaining Pre-trained Transformer Model

'Huggingface' is a an open-source platform directly accessible using 'transformers', a Python based library which allows download and implementation of pre-trained language models to be used for various tasks. 'Simpletransformers' is another Python based library which is a wrapper around the above-mentioned 'transformers' library and can be used to train and evaluate the models with ease. The library allows faster model initiation, training, and evaluation reducing total time taken. It further provides an easy-to-understand documentation, and automatically saves best model checkpoints with updated weights. Simpletransformers library utilizes 'Pytorch', a popular machine learning framework to encode and decode texts. PyTorch can handle models with greater speed.

4.7 Fine-Tuning the Model

Following successful download of the pre-trained model, the next step involves fine-tuning it on the prepared corpus. Fine-tuning a pre-trained model is more efficient than training a language model from scratch. During training, many model hyperparameters can be set to ensure proper generalization of the model.

4.8 Generating Text using the Model

The fine-tuned model can be used to generate text on unseen data. The output will depend on the specific task that the model was trained for. For example, if the model was trained for text summarization, the output will be a series of generated

summaries. In NLP applications, transformer-based language models are robust and produce high quality texts when appropriately fine-tuned.

4.9 Post-processing the Generated Text

The generated text require further processing to be accurately utilized in the evaluation stage. Post-processing steps include tokenization, removal of stop words, and lemmatization.

4.9.1 Tokenization

Tokenization refers to the process of segmenting sequences in words. Given a character sequence and a document defined unit, the words are split into 'tokens' which are stored separately. In some cases, the punctuation are also removed to better tokenize sequences. The model learns contextual representation of words from these tokens. There are different types of tokenizers available based on the requirements of the task to accomplish.

4.9.2 Stopword Removal

In most cases, there exists some words which do not add any relevant meaning to the sequence. These words are used to connect the words in a sequence but have no meaning on its own. It is important to remove these words to achieve accurate results. Words like 'a', 'an', 'the' are some examples of stopwords. In some NLP tasks involving text generation, stopword removal can decrease model performance. Hence, it is crucial to understand the task requirements.

4.9.3 Lemmatization

Lemmatization is a popular technique used for language modelling tasks. The process converts the words into its base form with a context. The same word may have multiple Lemmas. In this case, it is important to identify the Parts of Speech (POS) tag for the word in the specific context. For example, if the word ‘bothers’ is lemmatized on a noun context, it will return ‘bother’. Lemmatization does not make the words lose its contextual meaning which provides crucial information in language modelling tasks. Figure 21 shows versions of a word lemmatized to a noun context.

No.	original_word	lemmatized_word
1	bother	bother
2	bothered	bother
3	bothering	bother
4	bothers	bother

Figure 21. Lemmatization

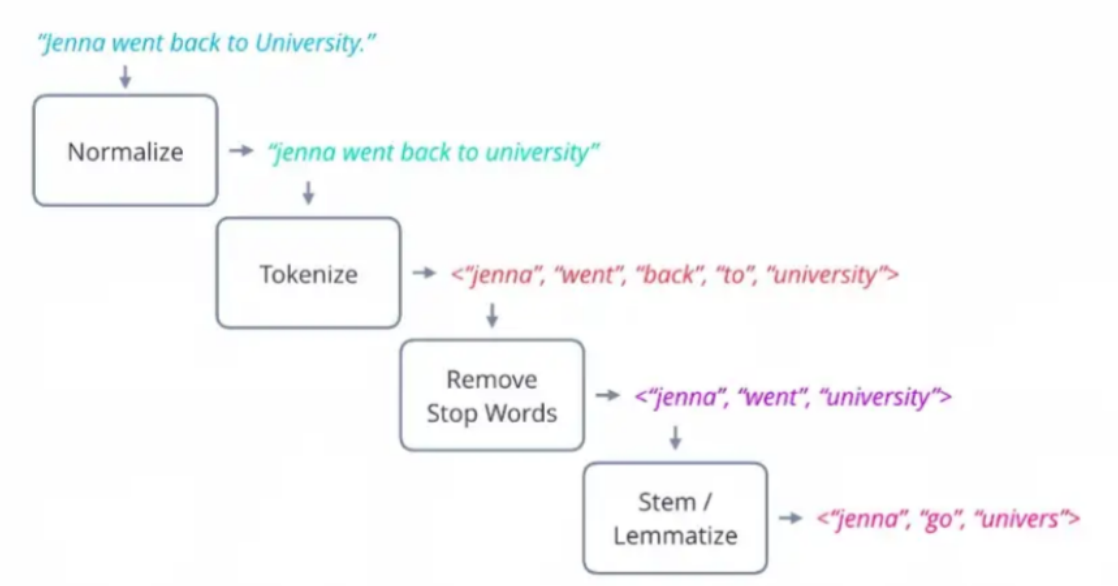


Figure 22. Steps of Data Post-processing (Pramod, 2020)

4.10 Measure extent of document relevance using Semantic Similarity Metrics

The degree of document relevance can be measured by calculating semantic similarity between generated text to targeted text. Commonly used evaluation measures for NLG tasks are BLEU Score (Papineni et al., 2002) and ROUGE Score (Lin, Chin-Yew. 2004). BLEU and Rouge are both standalone packages which can be used to measure similarity score between sequences. These metrics only measure n-grams of word similarity not taking word relatedness into account. Hence, these scores fail when applied to multi-disciplinary domains. Other methods of measuring semantic similarity between sequences of words may reflect the quality of modelling results more accurately.

Knowledge based semantic similarity measures rely on ontological representation of words. The structure of ontologies not only allows comparison between words but also their hyponyms, meronyms, and synonyms. 'Wordnet' (Miller, 1978) is such an ontology with a tree like structure between words where each child word can be traced back to its parent word. There exist metrics based on Wordnet calculates similarity of generated text based on the path length, information content and a combination of similarity and differences between two concepts. It calculates relatedness by considering the depths of the two words in the WordNet taxonomies. These metrics can be used in similarity evaluation using the 'Sematch' package (Zhu, G., Iglesias, C.A., 2017) in Python.

The transformer-based models have their own unique input encoding mechanisms which are crucial to finding words pairs with similar meaning or relatedness. The same mechanism can be applied to calculate semantic similarity between words

and concepts. SentenceTransformer, an open-source python package which utilizes pre-trained transformer-based models was included in the study, where a pre-trained RoBERTa model was used to calculate semantic similarity between the texts.

Lastly, two promising word embedding based models Word2Vec (Mikolov et al., 2013) and FastText (Bojanowski et al., 2016) can be used to measure semantic similarity between texts. The vectors in these models are generated in such a way that two semantically similar words exist close to one another in a vector space. The similarity between the words are measured using cosine similarity. FastText model is an extention of Word2Vec and operates at a more granular level with character n-grams where words are represented by the sum of the character n-gram vectors. These metrics can be accessed via the 'Gensim' package.

4.11 Analyze Results

The obtained results can be further analyzed to determine the eligibility for application of the acquired transformer-based models in the selected domain. Furthermore, the most informative semantic similarity measure can be determined for evaluation of generated text in the area. The graphical visualization of the fine-tuning process can also be obtained which will simplify interpretation of the results.

5 Computational Experiments

5.1 NLG Task Selection

There exists numerous articles and research publications on the Sustainability domain that discusses key issues and proposed techniques. Scientific publications are usually lengthy. Reading these papers and processing of these information gathered from them are time-consuming. Hence, there may be cases where key points are missed which may prove to be useful in taking a step forward towards the sustainable goals. Text summarization would prove to be a useful NLP task in such a scenario where a span of text would be summarized with key points and techniques. Usually, the abstract section of a publication consists of important key points illustrated in the publication. Abstractive summarization would not only provide crucial insights for the selected domain but also help determine primary issues and complexity in SDGs related research.

5.2 Corpus Development

Transfer learning requires fine-tuning complex language models on a representative corpus of documents from the selected domain. Scopus, Elsevier database covers scientific publications within natural science, social science and humanities. Therefore, it was considered a good source for collecting publications relating to the selected domain. The acquired corpus consisted of 21,692 scientific publications from January 2011 to September 2022. Information such as publication title, abstracts, keywords along with their metadata were extracted. To ensure that the most relevant publications were extracted, filtering with accurate keywords is a necessary step. Hence, the following keywords were used to search for scientific

publications in the area of sustainability -

(1) ”Sustainability”, (2) ”Sustainable development”, (3) ”Sustainable development goals”, (4) ”Environmental Sustainability”, (5) ”Green Economy”, (6) ”SDG Targets”, (7) ”SDG Indicators”, (8) ”Environmental Performance”, and (9) ”SDG Interlinkage.”

The ’Simpletransformers’ package necessitated a specific format for the data points. Each of training, evaluation and prediction sets included ”input-text” and ”target-text” pairs. It was observed that the abstract section of scientific publications were short and could be used as a great source for textual data as the ”input-text” to the model. Among the total number of extracted publications, 1082 publications did not include abstracts. At the data cleaning step, they were discarded from the corpus. Furthermore, the titles of these publications were used as the ”target-text” which was the summarized version of the input. The model fine-tuning was done on the corpus of 20,610 abstract-title pairs acquired from the selected publications.

Table 1. Examples of required Data Format for Fine-Tuning Process

Input text	Target text
Tikal has long been viewed as one of the leading polities of the ancient Maya realm, yet how the city was able to maintain its substantial population in the midst of a tropical forest environment has been a topic of unresolved debate among researchers for decades. We present ecological, paleoethnobotanical, hydraulic, remote sensing, edaphic, and isotopic evidence that reveals how the Late Classic Maya at Tikal practiced intensive forms of agriculture including irrigation, terrace construction, arboriculture, household gardens, and short fallow swidden coupled with carefully controlled agroforestry and a complex system of water retention and redistribution.	Forests, fields, and the edge of sustainability at the ancient Maya city of Tikal.
Several studies in the Anglo-American context have indicated that managers present themselves as morally neutral employees who act only in the best interest of the company by employing objective skills. The reluctance of managers to use moral arguments in business is further accentuated in the now common argument presented as a neutral fact that the company must always prioritise shareholder value. These and other commercial aims are seen as an objective reality in business, whilst questions about sustainability, environmental problems or fair trade are seen as emotional or moral ones; a phenomenon described as 'moral muteness'. This research explores whether this moral muteness is an Anglo-American phenomenon and/or whether managers in other countries - in this case Germany - might express themselves in a different way.	The moral muteness of managers: An Anglo-American phenomenon. German and British managers and their moral reasoning about environmental sustainability in business.

5.3 Preliminary Data Analysis

Bibliometric analysis is a scientific computer-assisted review methodology that can identify core research or authors, as well as their relationship, by covering all the publications related to a given topic or field (Han & Kim, 2020). The analysis can help to recognize research questions and their motivations using publication metadata. The analysis aids in understanding of acquired data and interpretation of the results. The analysis relies on the quality of data hence, data preparation steps such as dealing with missing values and removal of noise in data are crucial pre-requisites. This was accomplished using several python based libraries such as Pandas and NumPy.

Bibliometrics uses metadata from publications to identify thematic trends. It can be used as an indication of the importance and impact of the work or that of a research group, and therefore of its value to the wider research community. They further provide insight on the geographical nature of the collected data and determine progression of proposed strategies and tools. The investigation has been conducted to understand the distribution of areas among the collected publications. Choosing the appropriate methods for the bibliometric analysis is crucial.

Evolution of publications show that fewer number of publications were extracted for the year 2011. However, it started to gradually increase until the year 2020 and then it started to decline. Figure 23 was created using the 'Matplotlib' library and represents paper distribution per year.

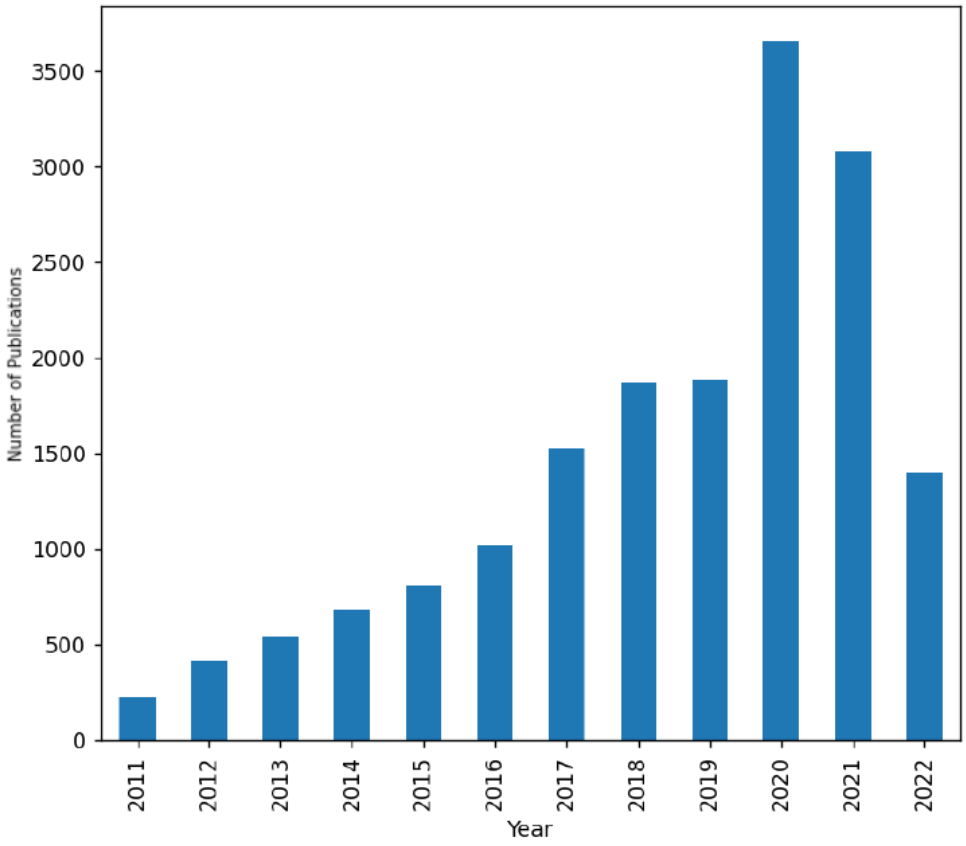


Figure 23. Distribution of Publications over the years

Despite the decline in the number of publications in the year 2021 and 2022, it is clear that research on sustainable development has gained popularity.

The distribution of citations per year is presented in Figure 24. It was generated using Tableau, a visualization tool.

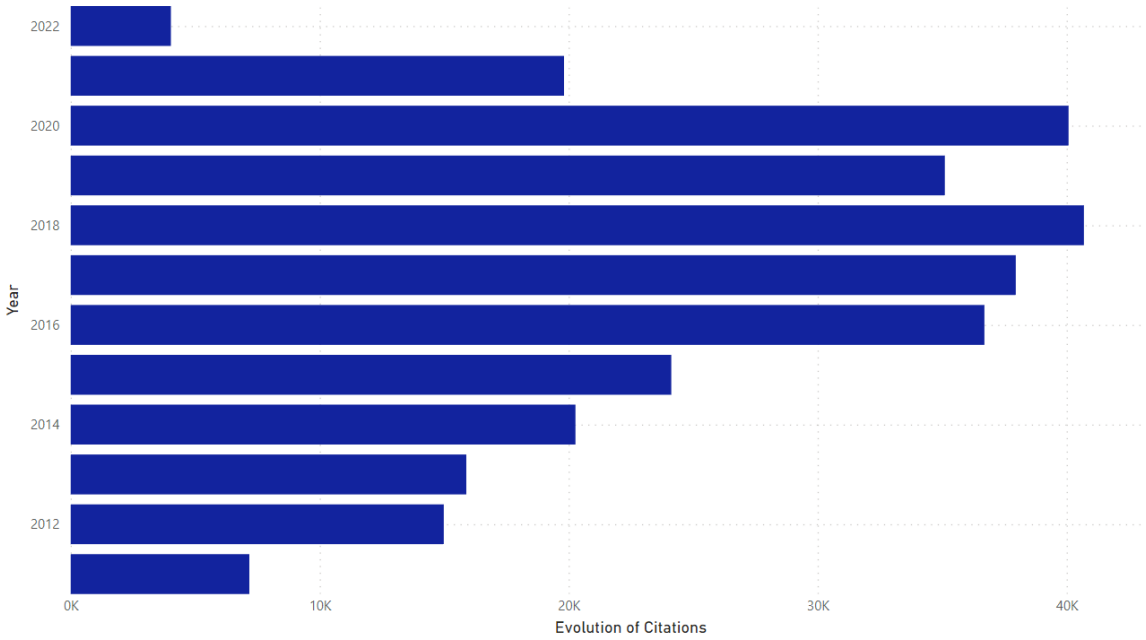


Figure 24. Distribution of Citations over the years

The increase in citation numbers also confirms the interest to the topics of sustainability. Decrease in citations in the recent years can be explained by the fact that citations take time. Reliable research that has been published in the present will be in great demand in the future.

Some sources of publications were more active than others. Figure 25 shows the sources with the highest number of publications in the corpus.

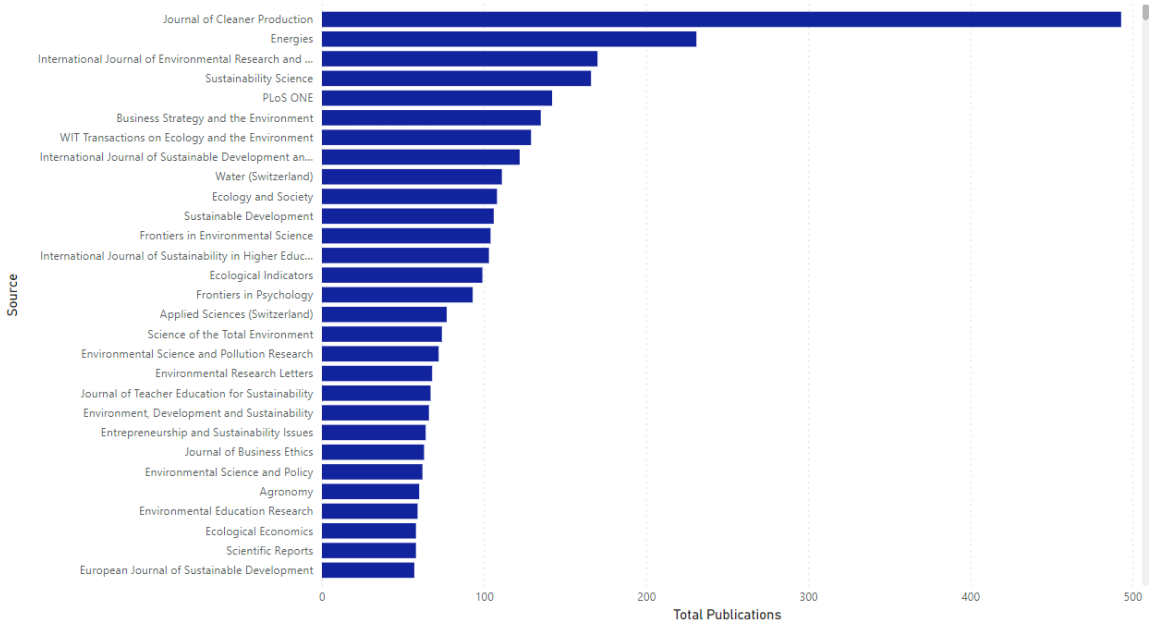


Figure 25. Distribution of top sources in the corpus

'Journal of Cleaner Production' provides around 500 publications, whereas the next popular source 'Energy' shows 224 publications. This indicates that the scopes of these two journals were of great interest in the research community.

Word clouds can give some insight into the themes and issues that are most present in a text. As such, they should be used in conjunction with other text analysis methods. N-grams of words can provide some contextual information about the existing data. The cloud was generated using the 'wordcloud' library in python.

Figure 26. Word Cloud of Most Frequent Words

Figure 26 shows the most frequent words in the corpus. Word bigrams and trigrams were also considered when implementing the analysis. 'Sustainability' and 'Sustainable Development' are the two most common word n-grams that appear in the corpus. Other noticeable word n-grams include 'sustainability indicator', 'development goal', 'renewable energy', 'climate change' and 'corporate sustainability'. These terms are key points that are utilized in majority of research incorporating environmental sustainability.

Table 2 presents the top 35 frequent words in the corpus.

Table 2. Frequency of Most Used Keywords

Keyword	Frequency
sustainability	11455
sustainable development	4918
education	2362
social	2344
environmental	2233
management	2009
energy	1703
assessment	1486
sustainable development goals	1163
governance	778
tourism	727
climate change	692
economy	685
ecological	537
financial	520
higher education	514
environmental sustainability	508
supply chain	488
sustainability assessment	486
agriculture	457
technology	445
evaluation	439
global	414
social responsibility	412
waste	409
land	405
consumption	399
social sustainability	388
covid 19	378
renewable energy	376
ecosystem	370
local	366
cycle assessment	366
life-cycle assessment	366
process	362

N-grams of words such as 'Sustainability' and 'Sustainable Development' appear in majority of the acquired publications and are used frequently. Words such as 'environmental', 'social', 'educational', 'climate change', 'higher education', 'renewable energy' and 'environmental sustainability' are also areas of research in the domain. These words confirm that the documents in the corpus relate to the different areas of sustainability.

Given that sustainability issues are global and require global efforts to combat them, understanding authors collaborative work is important for correct interpretation of the results. A ”Co-authorship network” is made up of the cooperation between two or more authors that are documented as collaborating on a study. The nodes are authors, who are connected by a line if they have co-authored one or more articles.

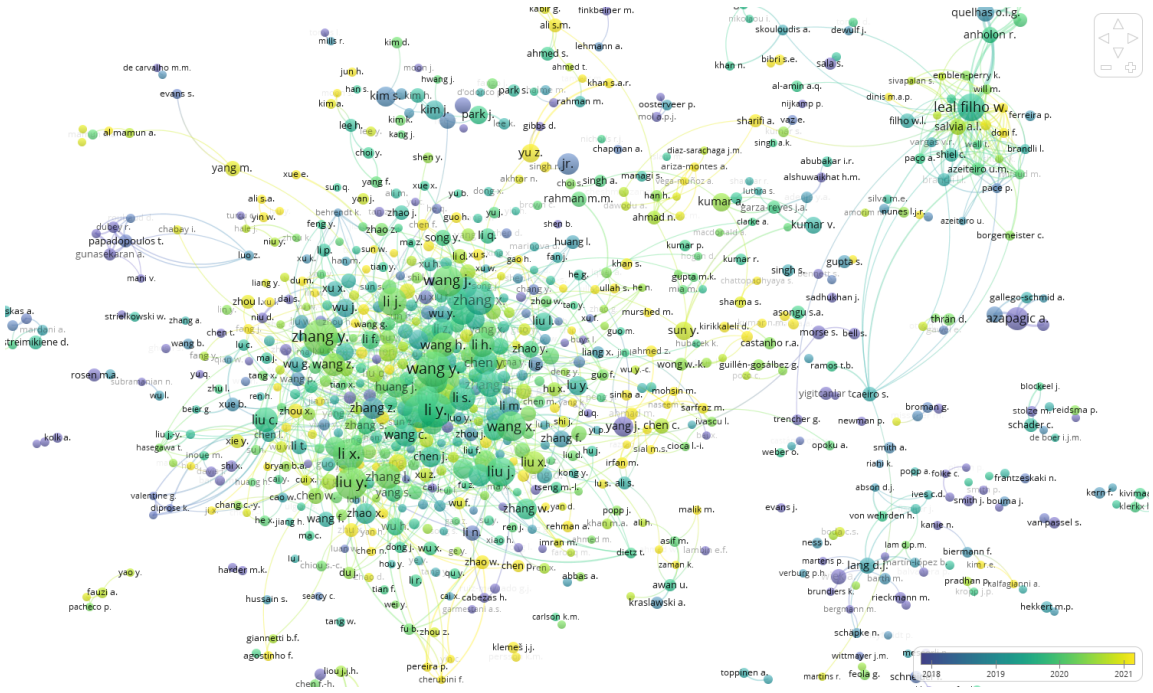


Figure 27. Co-authorship Overlay

Figure 27 indicates the co-authorship overlay and shows authors who have collaborated together on multiple studies in the domain of Sustainable Development Goals. The analysis represent a few authors namely Liu. Y., Wang. J., Zhang. Y., and Wang J. who have more collaborations in comparison to other authors in the field. The author Leal Filho, W. has also had multiple partnership with authors from Europe.

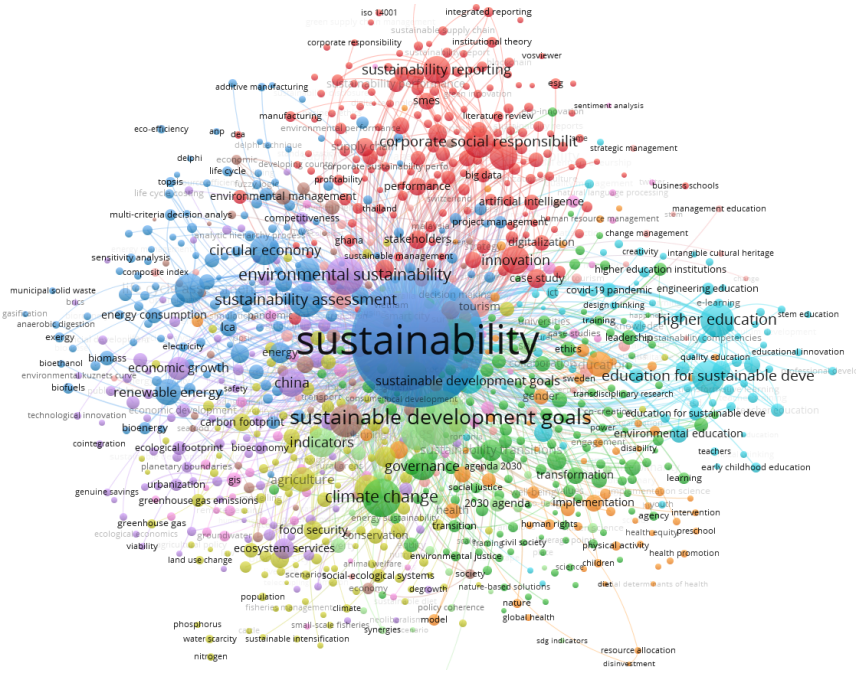


Figure 28. Keyword Co-occurrence network

A network of terms that occur together frequently can be seen in Figure 28. It shows multiple keyword clusters which are based on words that appear in a particular context in majority of the cases. For examples, n-grams 'renewable energy' and 'energy consumption' can be noticed as the most prominent keywords in the "blue" cluster which are often appear together. Other combinations include 'sustainability reporting' and 'social responsibility', 'agriculture' and 'food security', 'stakeholders' and 'innovation etc. The most recent trend within this field focuses on various industries, school systems, agriculture practices, food security, land and water practices, water resource management, poverty etc. The network was developed using the VosViewer software.

Thorough data exploration allowed to determine the most important aspects of the acquired corpus. The analysis confirms that the corpus represents the domain well and is suitable for use in fine-tuning process of the models.

5.4 Model Selection

A preliminary study was conducted using seven different types of transformer-based models using a corpus of scientific publications collected from Scopus, however there was no pre-determined domain. The study utilized the discussed fine-tuning approach to determine model performance on 3 tasks - text summarization, generation and machine translation. The common NLG metric - BLEU Score was used to measure semantic similarity between the generated text and the truth text.

Observing the results obtained, it was realized that encoder only models such as BERT and RoBERTa were not as effective as models that include a decoder component. These two models also did not perform well on the evaluation metric when compared to other models. Hence, they were not included in the experiment. There are other models that showed state-of-the-art performance in generating a span of text. GPT-2 model was also considered. However, due to extremely large number of model parameters, its fine-tuning process is very computationally expensive. For that reason this model was also not included in the study.

XLNET reportedly achieved state-of-the-art results on several NLP benchmark datasets. Model's permutation based mechanism is not always suitable for sequential data, where the order is important. Furthermore, XLNet is computationally more complex and processes all the tokens in the input sequence. This makes training slower and more memory-intensive.

Two language models showed higher scores on the evaluation metric for text summarization. They were BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation and T5: Text-to-Text Transformer. Therefore, these

two models were used as primary models in the experiments. The large version of these models consisted of 400+ million parameters and were obtained from the 'Huggingface' platform and used in the analysis.

5.5 Hyperparameters Setup

Simpletransformers library is pre-built with default hyperparameters which are used in training, however, these can also be manually set and are saved as a part of the learned model.

It is crucial to have the accurate hyperparameter values which ensure optimized performance. However, optimal model hyperparameters are not always known. Therefore, experimentation with these are key to ensure accurate usage.

In this study, certain model hyperparameters were set according to the summarization task requirement. Some of these help in tracking training progress, others aid in evaluation of the model on the test data. There are other hyperparameters that aid in illustrating how the model generalises in comparison to training data. Other default settings made by the library remained unaltered.

The same set of hyperparameter values were used for both BART and T5 models to realize the difference in model performance. The description of hyperparameters presented in Table 3 is available in the Appendix A.

Table 3. Values of Model Hyperparameters used in the Experiments

Hyperparameter Name	**Value**
do_sample	True
early_stopping_metric	evaluation_loss
evaluation_batch_size	16
evaluate_during_training	True
evaluate_during_training_steps	2500
evaluate_during_training_verbose	True
fp16	False
learning_rate	$4e$-5
max_sequence_length	128
number_beams	None
number_training_epochs	5
optimizer	AdamW
overwrite_output_directory	True
polynomial_decay_schedule_lr_end	$1e$-7
reprocess_input_data	True
save_evaluation_checkpoints	True
save_steps	-1
top_k	50
top_p	0.95
training_batch_size	8
use_early_stopping	True
use_multiprocessing	False
wandb_project_name	Training Visualizations
warmup_ratio	0.06
warmup_steps	0

5.6 Results

5.6.1 Fine-Tuning and Evaluating the Models

In addition to having superior empirical performance, pre-trained transformer-based models can be trained much quicker than recurrent or convolutional layer-based architectures. The training time however, differs between the type of a model as each of these models have their unique parameter settings and weights. Model size is another vital factor that determines training time; models with large number of parameters and dimensions take longer to train. The majority of transformer-based models come in a variety of sizes, and their application relies on the specified problem. Retraining a model enables it to make the most accurate predictions. This process does not change the parameters and variables used but adapts the model to the current data so that the existing parameters predict up-to-date outputs.

After each training epoch, a model checkpoint can be saved. Determining the fundamental reasons for poor model accuracy requires a thorough understanding of model fit. A model is said to be overfitting when it learns the information and noise in the training data to the point where it adversely affects the model's performance on new and unseen data. This indicates that the model learns concepts from the noise or random oscillations. These ideas don't apply to new data, which poses a problem for the model's ability to generalise. Underfitting refers to a model that can neither learn the training data nor generalize to new data. An underfit model is not suitable and shows poor performance. Therefore, a good balance between overfitting and underfitting is desired (Brownlee, 2019).

The study adopts a technique, where model performance is evaluated on data

unseen on the training step. By comparing the prediction error on the training and the testing data, it can be determined whether a predictive model is underfitting or overfitting. In summary, if the training loss is closer to the testing loss, the model is learning well and should be able to generalize accurately.

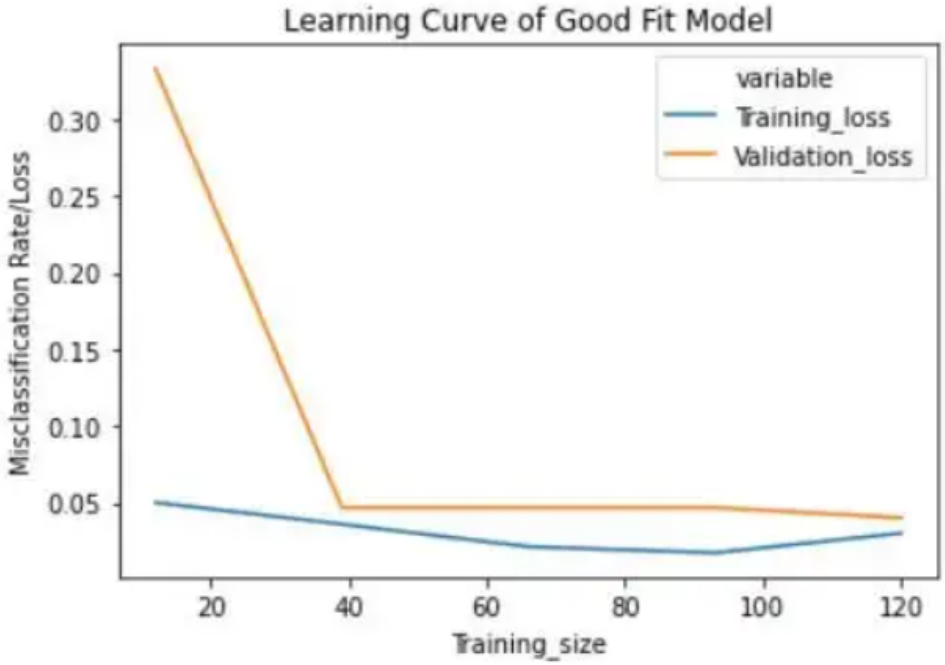
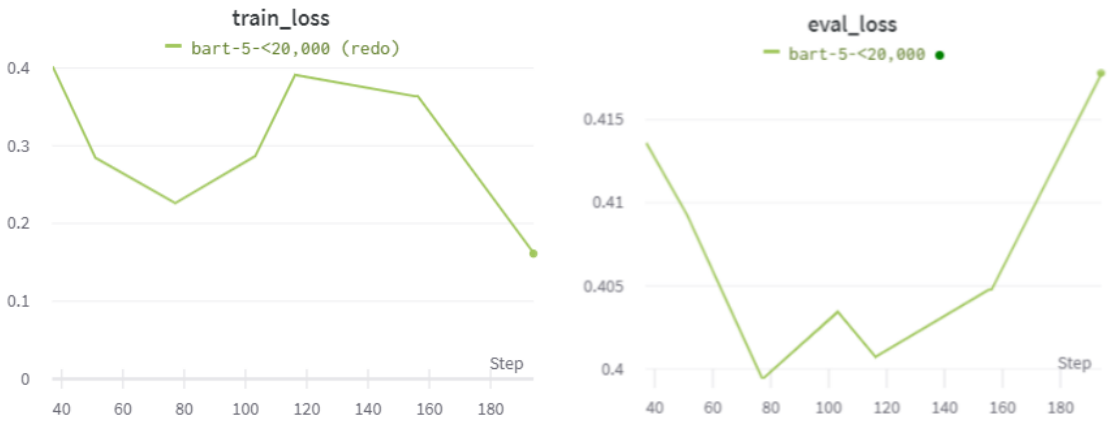


Figure 29. Learning Curves of a Good Fit Model (Muralidhar, 2021)

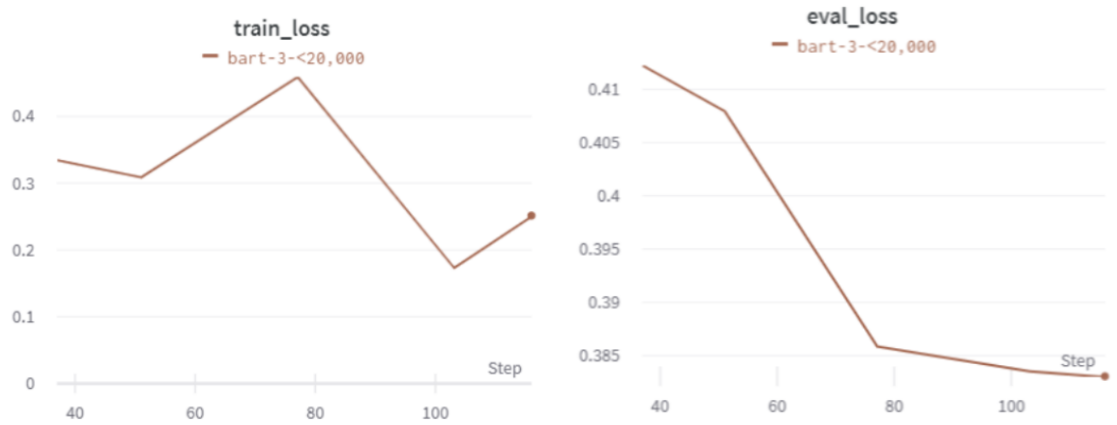
Initially, the BART and T5 models were set to run for 8 epochs. However, after first instance of training BART was completed, the model did not show a good example of a good-fit. This refers to the fact that both training and evaluation loss were quite far from each other. Further experimentation was conducted using lower number of epochs. Figure 30 shows BART's loss visualizations on 5 epochs.



(a) Training Loss (b) Evaluation Loss

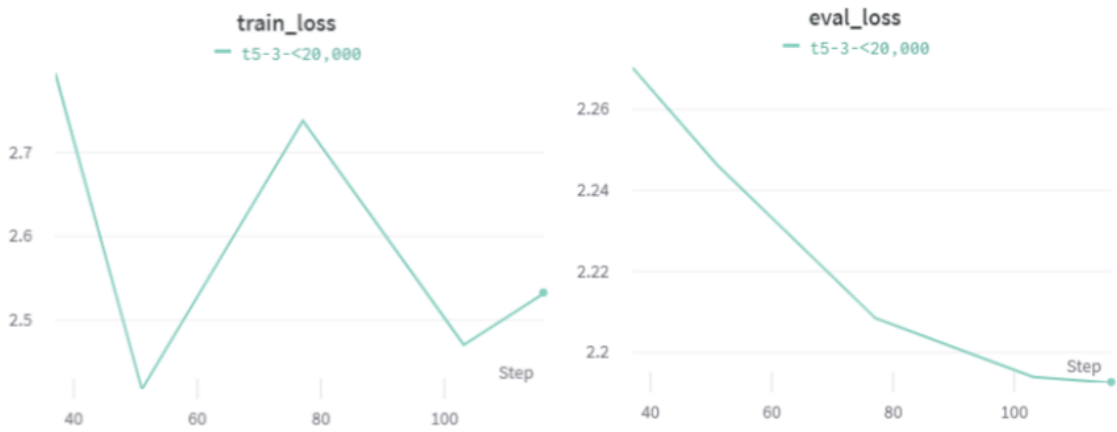
Figure 30. Dynamics of Loss Function of BART on 5 Epochs

Even though lower number of epochs were used, the difference between the losses were still quite significant. Therefore, even smaller number of epochs were required to be used in the next iteration. To generalize the issue, a regularization metric called 'Early Stopping' can be used. Such methods update the model so as to make it better fit the training data with each iteration. Early stopping rules provide guidance as to how many iterations can be run before the model begins to over-fit (Prechelt, 2012). This improves the model's performance on data outside of the training set. In conclusion, early stopping halts training when the metric no longer detects significant changes in the evaluation loss.



(a) Training Loss (b) Evaluation Loss

Figure 31. Dynamics of Loss Function (BART)



(a) Training Loss (b) Evaluation Loss

Figure 32. Dynamics of Loss Function (T5)

Figure 31 and 32 show a better representation of good fit models. These fine-tuned instances of BART and T5 were then used to derive summaries using the unseen prediction set.

5.6.2 Qualitative Analysis of Model Generated Text

Although some of the top layers are unfrozen and re-trained during fine-tuning, majority of language structures that the model learns is from the pre-training phase. Transformers, as opposed to RNNs, have the advantage of learning to replace words with appropriate equivalents or words that are linked to the context. It was claimed that BART and T5 generate text that cannot be distinguished from human written texts. In the experiments, the models were fine-tuned in such a way that they would not show numerical values that represent results obtained by the research but would only highlight the most important keywords that represent techniques used and areas explored.

Table 4 indicates that both models generate text that succinctly summarise the input text's main ideas and, in some cases, produce terms that are synonyms for or related to a term in the input text, but better characterise the text description. However, BART creates summaries that closely resemble the original text and generates representative keywords, whereas T5 typically creates text that omits these in-depth terms. In the third example - economy, society, and environment are the "Three pillars of sustainability in Canada," according to BART, while T5 simply makes reference to the three pillars. After conducting qualitative comparison between the text-to-predict and the summaries, it is evident that in majority of the cases, summaries generated from the models are more applicable for inclusion in the analysis phase than the original publication title, because they provide a much more concise and accurate representation of the study. In certain instances, they are a paraphrased version of the original writings.

Table 4. Qualitative Comparison of samples of Model-Generated Summaries with Original Publication Titles

Text to predict	BART Summary	T5 Summary
Food security and sustainability: Exploring Japanese youth participation in agriculture.	Youth engagement in the Japanese agriculture industry: A qualitative study on sustainability of food supply.	Youth involvement in food sustainability in Japan: A phenomenological study.
Innovation: A Moderating Factor for the Effectiveness OF Diversification Strategies.	Innovation as a moderating factor for the effectiveness of diversification strategies on the sustainability of hospitality firms in Nigeria.	The moderating effect of innovative diversification strategies on sustainability of hospitality firms: A case study of Nigerian four-star hotels.
Canada's oil sands industry from a sustainability perspective.	Impacts of oil sands development on the economy, society and the environment as three pillars of sustainability in Canada.	Three pillars of sustainability: A study of the oil sands industry in Canada.
Analysis of scale factors on China's sustainable development efficiency based on three-stage DEA and a double threshold test.	Super-slack-based measure of China's sustainable development efficiency based on the driver-pressure-state-impact-response framework: A three-stage data envelope analysis.	Sustainable development efficiency: A three-stage data envelope analysis approach based on super-slack-based data analysis in China.
Analysis of OpenStreetMap Data Quality for Selected Counties in Poland in Terms of Sustainable Development.	Evaluation of spatial data quality elements of 'OpenStreetMap' for monitoring sustainable development goals	A Statistical Evaluation of the Spatial Data Quality Elements and Sustainable Development Indicators using OpenStreetMap.

5.6.3 Comparative Analysis of summaries generated by different models

The accuracy of the model generated text were initially evaluated by comparing model generated summaries against the publication title. However, it was anticipated that the two models would occasionally generate text that would be much more in-depth than the publication title itself i.e., the models may be predicting techniques and methods that may or may not exist in the title. Metadata from publications contain key-terms that outline important keywords explored in the article. In most cases, these keywords contain relevant techniques used, application areas, field of research and methodologies adopted. Because these keywords can gauge the relevance of the projected text depending on whether they appear in them or not, comparing generated summaries with the publication keywords would also be beneficial. Lastly, the generated summaries between BART and T5 could also be compared using the evaluation metrics to identify differences among the text generation between the models.

Summaries generated by Transformer-based models are robust and often use terms that may not be present in the original text but have relational meaning in the context. This is where majority of the NLG evaluation metrics fail because they only match words to determine similarity. Therefore, the experiment adopts different methods of calculating semantic similarity between words. These metrics have their unique method of calculating similarity and the difference between the scores helps to realize the applicable metric for use in the domain. Before the generated summaries can be used in comparison with the keywords, they were required to be post-processed. Table 5 outlines the scores on the metrics when comparing the generated text against the publication keywords.

Table 5. Semantic Similarity Scores between Generated Summaries and Publication Keywords

Model	**Semantic Similarity Measures**							
	BLEU	ROUGE	Transformer	WUP	RES	LIN	Word2Vec	FastText
BART	0.21	0.23	0.596	0.27	0.45	0.17	0.63	0.71
T5	0.20	0.18	0.599	0.26	0.42	0.13	0.61	0.69

Here, the terms WUP, RES, and LIN correspond to the Wu & Palmer, Resnik, and Lin Similarity metrics respectively.

The scores are depicted on a scale of 0 to 1. The first metric used in the comparison is BLEU Score and is considered one of the most popular similarity metrics to evaluate Natural Language Generation tasks. However, in this instance, BLEU-1, the Unigram Precision Score was used to compare the texts as the comparison was actually between two sets of keywords and not full-length texts. BLEU Score show low scores when used for comparison in the domain and only shows a score of 0.21 on BART generated summaries and 0.20 on T5 generated summaries. That was expected as transformer generated text do not follow the exact text structure and often predict words that are not present in the corpora but are meaningful and applicable to the particular context. These words may be synonyms, hyponyms and meronyms of words that are present in the original text to predict. Metrics such as BLEU Score do not consider different forms of these words and only rely on n-gram matching. This explains the low score describing performances of both the models.

Rouge Score is the second metric used in this study. It is similar to BLEU Score. However, BLEU measures Precision i.e., how many words or n-grams in the model generated summaries appear in the human reference summaries, while Rouge measures Recall i.e., how many words or n-grams in the human reference sum-

maries appear in the machine generated summaries. Rouge Score is also commonly used to evaluate NLG tasks alongside BLEU. It does not appreciate robust text generation as it also relies on overlapping of unigram between the hypothesis and reference summaries. Rouge also performs poorly with a score of 0.23 in BART generated summaries and 0.19 in T5 generated summaries.

The analysis of traditional NLG evaluation metrics called for exploration of other methods that can be applied to text generation by seq2seq transformers such as BART and T5 in a multi-disciplinary domain. In this regard, similarity metrics based on large lexical database of English words known as Wordnet can be explored. The study utilizes three types of semantic similarity metric based on Wordnet: (1) Wu & Palmer Similarity, which measures similarity depending on the path distance between two words; (2) Resnik Similarity, which compares the information content between two words and (3) Lin Similarity assigns a score based on the combination of commonality and differences between the words. Wu & Palmer (WUP) calculates the similarity based on how similar the word senses are and where the words occur relative to each other in the hypernym tree. However, given the large structure of Wordnet, this metric is not very appropriate for measuring semantic similarity in the particular problem as the calculation of path distance between sub areas of sustainability may be quite large. Scores of 0.27 and 0.26 on BART and T5 summaries suggest that these metrics perform better than the NLG evaluation metrics but are not very informative for the domain. Reznik metric shows better results; 0.45 and 0.42 on BART and T5 summaries respectively with comparison to WUP as the metric compares the information content rather than path length between words. Lin Similarity compares information content of words projecting a score on a combination of commonality and differences. This metric

scores less than 0.20 for both model generated summaries. Perhaps, the differences between the words create a bottleneck when measuring similarity, which shows lower score. Therefore, among the three metrics that utilize the Wordnet ontology, it can be concluded that RES Similarity provides the best result.

Sentence Transformer calculates similarity between sentences using a pre-trained deep neural network. The Transformer-based model - RoBERTa was used in this regard. RoBERTa uses Byte Pair Encoding (BPE) to encode text which has a vocabulary of 30,000 words. These embeddings capture the semantic meaning and context of the inputs and can be compared to determine their similarity. The similarity score is then calculated as the cosine similarity between the embeddings. This metric outperforms Wordnet based metrics and the NLG metrics with a score of 0.60 for both experimented models. SentenceTransformer's similarity performance is likely due to several factors, including the use of state-of-the-art transformer-based models for encoding sentences, fine-tuning on large datasets, and the ability to capture context and semantic relationships between words in a sentence. The pre-trained models have also been trained on a diverse range of tasks. This may contribute to their overall performance on a variety of text-related tasks, including sentence similarity.

Word2Vec model scores 0.63 on BART summaries and 0.61 on T5 summaries whereas FastText, an extention of Word2Vec scores 0.71 and 0.69 respectively. These models capture the semantic relationships between words in a continuous dense vector space which allows for computing the similarity between words or sentences by using simple distance measures, such as cosine similarity, on their vector representations. In Word2Vec, the vector representation of a word is trained to predict its surrounding context words in a corpus, capturing the semantic and

syntactic information of the word. This representation can capture the meaning of the word and its relationship with other words. FastText extends Word2Vec by also considering sub-word information, allowing it to handle out-of-vocabulary words and variations of words. The resulting vector representations capture the meaning of words and their relationships with other words, as well as the relationships between sub-words, making it a more robust model for measuring text similarity. Overall, Word2Vec and FastText provide a simple yet effective way to measure the semantic similarity between words or by mapping them to a continuous dense vector space, where the similarity can be computed using distance measures.

Furthermore, Table 6 lists the scores achieved when comparing generated summaries to the publication titles.

Table 6. Semantic Similarity Scores between Generated Summaries and Publication Titles

Model	**Semantic Similarity Measures**							
	BLEU	ROUGE	Transformer	WUP	RES	LIN	Word2Vec	FastText
BART	0.24	0.41	0.69	0.26	0.66	0.135	0.66	0.74
T5	0.20	0.27	0.65	0.25	0.67	0.133	0.63	0.72

This comparison does not yield massive differences within majority of the metrics. Certain measures, however, may perform better when comparing sequences rather than groups of terms, thus they may see some performance gains. A noticeable difference can be seen in the ROUGE metric with a score of 0.49 however, it is still not significant in comparison to the other metrics. Bigram Precision Score of BLEU retains similar scores in this regard. When comparing sequences rather than keywords, the SentenceTransformer-based metric performs better with a boost of 10%. The measurements based on Word2Vec are the most reliable and display good results in both comparisons.

Lastly, Table 7 compares the summaries generated by BART and T5 with each other to realize the similarities or differences in text generation among the models.

Table 7. Semantic Similarity Scores between Generated Summaries by BART and T5

Model	**Semantic Similarity Measures**							
	BLEU	ROUGE	Transformer	WUP	RES	LIN	Word2Vec	FastText
BART & T5	0.40	0.12	0.71	0.26	0.65	0.14	0.71	0.78

Within the first two stages of comparative analysis, BART model outperformed T5 on majority of the semantic similarity metrics. It was further discovered that the model generated summaries were thorough and produced significant keywords that were pertinent to the study. Hence, it can be argued that BART is a suitable seq2seq model for use in applications relating to sustainability. Furthermore, Word2Vec based models are the most consistent in the explored comparisons and show the best results. Therefore, this metric should be widely used to evaluate transformer-based generations in the area.

5.7 Analyzing SDGs Representation on Generated Summaries

The proposed framework was used to determine the degree of relevance of acquired scientific publications to the 17 SDGs. The framework was applied to a corpus of recent publications on the SDGs topics indexed in Scopus, Elsevier database. The fine-tuned BART model from the previous steps was used in the analysis. This time scientific publications were extracted from Scopus using the only keyword 'Sustainable Development Goal'. This specific keyword was used to identify publications that explicitly describe SDGs and their attainment. The keyword was searched within author keywords, publication abstracts and titles to ensure all relevant publications can be acquired. The new corpus consisted of 988 publications marked with the appearance date between January 2022 and February 2023.

The abstracts from the collected publications were processed following the proposed framework and used as input to the model to obtain summaries. The post-processed summaries were then compared to the descriptions of each of the 17 SDGs using the Word2Vec similarity measure so that for each publication from the corpus its relevance to each SDG is calculated. The analysis allowing for identification of trends in research on SDGs.

The descriptions of SDGs were downloaded from the official UN website. They are presented in Table 8.

Table 8. Description of the SDGs

Goals	Description
SDG 1	End poverty in all its forms everywhere
SDG 2	End hunger, achieve food security, improved nutrition and promote sustainable agriculture
SDG 3	Ensure healthy lives and promote well being for all at all ages
SDG 4	Ensure inclusive and quality education for all and promote lifelong learning
SDG 5	Achieve gender equality and empower all women and girls
SDG 6	Ensure access to water and sanitation for all
SDG 7	Ensure access to affordable, reliable, sustainable and modern energy for all
SDG 8	Promote inclusive and sustainable economic growth employment and decent work for all
SDG 9	Build resilient infrastructure, promote sustainable industrialization and foster innovation
SDG 10	Reduce inequality within and among countries
SDG 11	Make cities inclusive, safe, resilient and sustainable
SDG 12	Ensure sustainable consumption and production patterns
SDG 13	Take urgent action to combat climate change and its impacts
SDG 14	Conserve and sustainably use the oceans, seas and marine resources
SDG 15	Sustainably manage forests, combat desertification halt, reverse land degradation and biodiversity loss halt
SDG 16	Promote just peaceful and inclusive societies
SDG 17	Revitalize the global partnership for sustainable development

Table 9 shows the minimum, maximum and the average scores for each SDG.

Table 9. Aggregate similarity scores of documents in the corpus to SDGs

Goals	Minimum Score on a summary	Maximum Score on a summary	Average
SDG 1	0.17	0.63	0.46
SDG 2	0.13	0.82	0.58
SDG 3	0.12	0.65	0.46
SDG 4	0.11	0.81	0.51
SDG 5	0.11	0.77	0.38
SDG 6	0.07	0.74	0.43
SDG 7	0.09	0.72	0.54
SDG 8	0.14	0.79	0.61
SDG 9	0.14	0.81	0.60
SDG 10	0.08	0.64	0.42
SDG 11	0.12	0.68	0.50
SDG 12	0.10	0.77	0.53
SDG 13	0.12	0.72	0.49
SDG 14	0.11	0.64	0.43
SDG 15	0.18	0.68	0.45
SDG 16	0.15	0.64	0.44
SDG 17	0.08	0.82	0.63

According to numerous publications on NLP similarity measures, scores above 0.65 indicate a high level of similarity. Scores above 0.5 are frequently viewed as positive, whereas those below 0.4 are interpreted negatively. The mean scores that each goal received in the corpus are shown in the 'Average' column. SDG 5 is the least represented goal in the corpus with a score of 0.38, followed by SDG 10, 14, 6 and 16 with scores just above 0.40. Under-representation of this important goal should be further investigated. Among the SDGs which are represented much stronger in the corpus are SDG 17 (0.63), SDG 8 (0.61) and SDG 9 (0.60). Majority of the goals that represent environmental studies have low scores. In terms of calculated averages, it is evident that the corpus represents the goals relating to economical development much clearly than those within social and en-

vironmental studies.

The obtained similarity scores can be further analyzed using an approach presented in (Erechtchoukova and Safwat, 2023). To further verify the representation of goals in the corpus, the maximum and minimum scores were also analyzed. It can be seen that the highest scores on a single summary were achieved by SDG 17 and SDG 2 with 0.82 each. However, these scores on a single summary may not be as informative as these publications may be fewer in number. In this regard, to better analyze the extreme scores, SDGs that were the most and least represented could be identified using similarity scores between the generated summaries and SDGs.

Table 10. Determining most and least relevant goals in the corpus

Goals	**No. of times the goal is least relevant to summaries**	**No. of times the goal is most relevant to summaries**
SDG 1	20	3
SDG 2	0	72
SDG 3	15	4
SDG 4	1	47
SDG 5	549	10
SDG 6	75	4
SDG 7	0	14
SDG 8	0	173
SDG 9	0	87
SDG 10	75	5
SDG 11	1	1
SDG 12	5	30
SDG 13	0	18
SDG 14	181	6
SDG 15	31	13
SDG 16	34	1
SDG 17	1	500

Table 10. illustrates the results achieved. SDG 5 had the least relevance to 549 publications in the corpus, followed by SDG 14, 10 and 6. SDG 17 is the most represented goal in the corpus. The next two SDGs that were most relevant to higher number of publications are 8 and 9. According to the findings, SDGs 17, 8, 9 and 2 were addressed in all the extracted publications. Therefore, these goals represent the contemporary research trend.

The analysis, however, is not bound by keywords in the description of the SDG. It can be seen that SDG 4 (most relevant to 47 publications) had neither the word 'Sustainable' nor 'Development' in the description, but has relevance to more publications than SDG 7 (most relevant to 14 publications) which had the word 'Sustainable' in the description.

It is also worth noting that a corpus acquired using different keywords would give different results i.e., it may give an alternate conclusion on the representation of SDGs. The proposed framework would still be applicable and support any further analysis to determine the degree of document relevance to any targeted text.

A single generated summary may represent multiple goals depending on the area of application. Pairwise correlation uncovers these potential relations of interests. In this regard, the correlation can be used to identify additional goals that are represented in summaries that correspond to a certain goal. Figure 33 shows the results obtained from the pairwise correlation test.

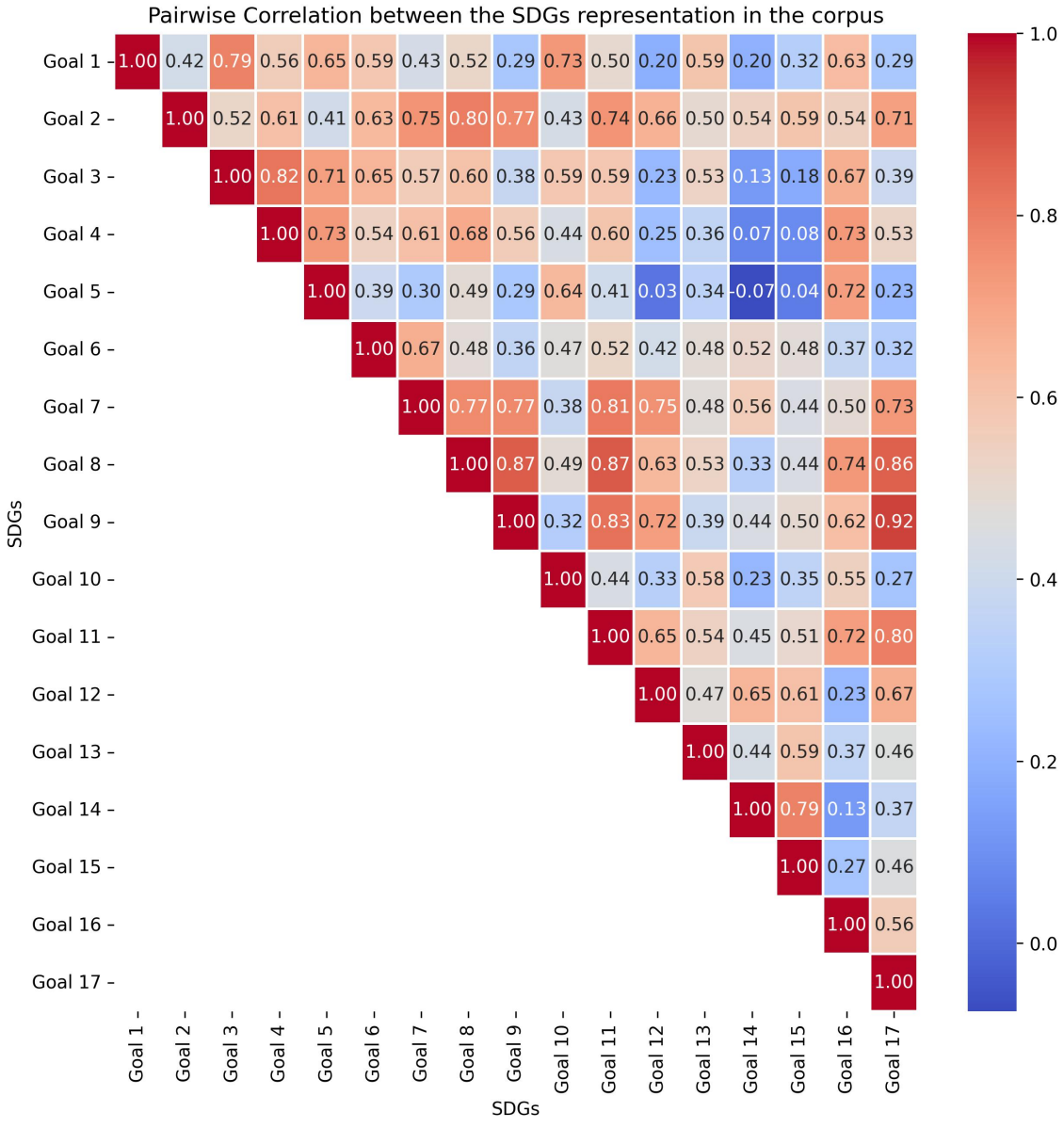


Figure 33. Heat map showing Pairwise Correlation results

The heat map shows the correlation co-efficient of SDGs across all the generated summaries. The strongest correlation was found among SDG 9 and SDG 17 with a score of 0.92. There are high correlations among SDGs 8 and 9, 8 and 11, 9 and 11, 10 and 11 with scores depicted above 0.80. Furthermore, there are also noticeable correlations among SDGs 1 and 3, 2 and 8, 11 and 17, 14 and 15. SDG 17 is the most explored area within recent publications. Among the goals that appear together the least are SDGs 5 and 4 which receive negative correlation scores. Others include SDG 5 and 15, 5 and 17. SDG 5 is the least represented in the corpus, the correlation of majority of SDGs with SDG 5 also show low scores. An interesting insight is the correlation between SDG 14 and SDG 15, with a score of 0.79. SDG 14 resembles sustainable oceans whereas SDG 15 discusses manageable forests. Even though these words have different semantic meaning, the Word2Vec based metric was able to find these crucial and unknown relationships among these areas to such extent. Furthermore, majority of the other scores among other goals show good correlations. Therefore, the pairwise correlation support the claim that majority of the goals are inter-dependant.

6 Conclusion

The main contribution of the study is a transfer learning framework that adopts semantic similarity analysis utilizing transformer-based language models and abstractive text summarization to determine the degree of relevance of documents by comparing them to targeted short texts in the specific domain. The study was conducted on a developed corpus of documents and included comparative analysis of transformer-based language models. Two popular seq2seq models, BART and T5, were fine-tuned and their performance was evaluated qualitatively, and using multiple semantic similarity measures. The proposed framework was applied to the corpus of publications on investigation and attainment of SDGs. This allowed to determine goal distribution in the corpus.

It is worth noting, that identification of trends and gaps in such multi-disciplinary domain as Sustainability is challenging. Application of NLP can streamline the analysis of unstructured text provided that a methodological framework is available. The proposed framework can be used to analyse any sub-sector within the sustainability umbrella. The textual data included an accurate representation of scientific publications from all the existing disciplines in the domain. This allowed the model to gain required knowledge to predict appropriate summaries based on unseen input data. Abstractive summarization is an NLP task that offers important details about the document in a concise form.

Transformer-based language models forming the core of the framework support its efficient implementation because they are non-sequential. Sentences are processed as a whole rather than word by word which makes them faster with longer contextual dependency compared to traditional RNN based models. However,

full-length publications are still not appropriate for the analysis as the contextual dependency comes with a fixed length. As a result, these models work best when short to medium-length sequences are injected.

Both the BART and T5 models' fine-tuned iterations demonstrated robustness in text generation. The generated summaries can be used instead of the original publication titles in the analysis as it produces a much more concise summary of the abstract. These fined-tuned checkpoints can be applied to other seq2seq tasks, and produce high-quality text as long as the input content is within the domain. The BART model however, outperformed the T5 model in majority of the semantic similarity measures. BART demonstrated superior scores over T5 on the NLG evaluation, Wordnet based, and Word embedding based metrics. T5 however, only showed higher scores on the Transformer-based metric. Hence, the BART model was used to analyze representation of SDGs in the corpus.

Experimentation using the semantic similarity measures showed critical differences. NLG evaluation metrics that are widely used in text generation resulted in very low scores as they only consider word-to-word matching. Therefore, selecting the appropriate semantic similarity measure is an important step in any modelling exercise. Among the measures, word-embedding based models which rely on representations of words in a vector space appeared to be more informative. Therefore, it can also be concluded that statistical models such as Word2vec are the most appropriate when measuring semantic similarity between sequences within the selected domain.

Further analysis on SDGs in the acquired corpus of recent publications exhibits the trend of research in the domain. Collaboration network among researchers (Figure 27) indicate that there are intensive ongoing efforts within research com-

munity in the domain. However, the discrepancies between the scores of SDG representation in the corpus show that some SDGs receive more attention than others. Areas that lack research should be explored further.

The study provides a useful framework that can be used in the sustainability domain that condenses a given text into its brief form while clearly indicating the topics covered and techniques suggested for pressing problems. With regards to analyzing representation of the SDGs in a given corpus, the study argues that the framework maps scientific publications to SDGs with much higher relevance than multi-class classification because semantic similarity metrics have the ability to capture complex and unknown relationships among the words. Furthermore, future research on attainability of SDGs may greatly benefit from the application of these unified frameworks because it allows to automatically process textual data and their semantic similarity to the SDGs.

7 References

Ali, S. & Shirazi, F. (2022) A Transformer-Based Machine Learning Approach for Sustainable E-Waste Management: A Comparative Policy Analysis between the Swiss and Canadian Systems. Sustainability, 14, 13220 https://doi.org/10.3390/su142013220

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

Anunaya, S. (2022) Data preprocessing in data mining - A hands on guide, Available at: https://www.analyticsvidhya.com/blog/2021/08/data-preprocessing-in-data-mining-a-hands-on-guide/ (Accessed: January 16, 2023).

Araci, D. (2019). FinBERT: Financial Sentiment Analysis with Pre-trained Language Models. Retrieved 15 June 2022, from https://arxiv.org/abs/1908.10063

Arslan, Y., Allix, K., Veiber, L., Lothritz, C., Bissyandé, T.F., Klein, J., & Goujon, A. (2021). A Comparison of Pre-Trained Language Models for Multi-Class Text Classification in the Financial Domain. Companion Proceedings of the Web Conference 2021.

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.

Bessou, S., & Chenni, G. (2021). Efficient Measuring of Readability to Improve Documents Accessibility for Arabic Language Learners.

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with sub-word information. Transactions of the association for computational

linguistics, 5, 135-146.

Bozinovski, S., & Fulgosi, A. (1976). ”The influence of pattern similarity and transfer learning upon the training of a base perceptron B2.” (original in Croatian) Proceedings of Symposium Informatica 3-121-5, Bled

Brinkmann, A. & Bizer, C. (2021). Retrieved 18 June 2022, from https://www.uni-mannheim.de/media/Einrichtungen/Brinkmann-Bizer-Improving Hierarchical Product Classification using domain specific language modelling-PKG4 E-commerce 2021

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., & Dhariwal, P. et al. (2020). Language Models are Few-Shot Learners. Retrieved 13 June 2022, from https://arxiv.org/abs/2005.14165

Brownlee, J. (2018). Retrieved 13 September 2022, from https://machinelearning mastery.com/encoder-decoder-recurrent-neural-network-models-neural-machine-translation

Brownlee, J. (2019) Overfitting and underfitting with machine learning algorithms, Available at: https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms/ (Accessed: January 29, 2023).

Charnes, A., Cooper, W. W., & Rhodes, E. (1978). Measuring the efficiency of decision making units. European journal of operational research, 2(6), 429-444.

Chen, B., Chen, B., Gao, D., Chen, Q., Huo, C., & Meng, X. et al. (2021). Transformer-based Language Model Fine-tuning Methods for COVID-19 Fake News Detection. Retrieved 15 June 2022, from https://arxiv.org/abs/2101.05509

Conforti, C., Hirmer, S., Morgan, D., Basaldella, M., & Or, Y. B. (2020, November

17). Natural language processing for achieving sustainable development: The case of neural labelling to enhance community profiling. arXiv.org. Retrieved October 8, 2022, from https://arxiv.org/abs/2004.12935

Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q., & Salakhutdinov, R. (2019). Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context. Retrieved 12 June 2022, from https://arxiv.org/abs/1901.02860

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Friederich, S. (2021) Fine-tuning, Stanford Encyclopedia of Philosophy. Stanford University. Available at: https://plato.stanford.edu/entries/fine-tuning/ (Accessed: January 16, 2023).

Donthu, N., Kumar, S., Mukherjee, D., Pandey, N., & Lim, W. (2021). How to conduct a bibliometric analysis: An overview and guidelines. Journal Of Business Research, 133, 285-296. doi: 10.1016/j.jbusres.2021.04.070

Duan, Y., Edwards, J. S.,& Dwivedi, Y. K. (2019). Artificial intelligence for decision making in the era of Big Data–evolution, challenges and research agenda. International Journal of Information Management, 48, 63–71.

Erechtchoukova, M.G., Safwat, N., 2023. Identifying document relevance to Sustainable Development Goals using NLG. In: Proc. MODSIM2023, 25th International Congress on Modelling and Simulation, Modelling and Simulation Society of Australia and New Zealand, July 2023 (forthcoming)

Gillioz, A., Casas, J., Mugellini, E., & Abou Khaled, O. (2020). Overview of the Transformer-based Models for NLP Tasks. In 2020 15th Conference on Computer

Science and Information Systems (FedCSIS) (pp. 179-183). IEEE.

Guisiano, J., Chiky, R. & Mello, J. (2022) SDG-Meter : a deep learning based tool for automatic text classification of the Sustainable Development Goals. ACIIDS :14th Asian Conference on Intelligent Information and Database Systems, Nov 2022, Ho Chi Minh, Vietnam. ffhal-03738404f

Hák, T., Janoušková, S., & Moldan, B. (2015, August 21). Sustainable development goals: A need for relevant indicators. Ecological Indicators. Retrieved October 8, 2022, from https://www.sciencedirect.com/science/article/pii/S1470160X15004240

Han, J., Kang, H., Kim, M., & Kwon, G. (2020). Mapping the intellectual structure of research on surgery with mixed reality: Bibliometric network analysis (2000–2019). Journal Of Biomedical Informatics, 109, 103516. doi: 10.1016/j.jbi.2020.103516

Harly, W., Kwee, H. R., & Suhartono, D. (2022). Quantitative Argument Summarization using Text to Text Transformer. ICIC Express Letters, Part B: Applications, 13(7), 749-756. doi:10.24507/icicelb.13.07.7494

Hochreiter, Sepp & Schmidhuber, Jürgen. (1997). Long Short-term Memory. Neural computation. 9. 1735-80. 10.1162/neco.1997.9.8.1735.

Kanani, B. (2019). Stemming and Lemmatization - Machine Learning Tutorials. Retrieved 28 August 2022, from https://studymachinelearning.com/stemming-and-lemmatization/

Kingma, D., & Ba, J. (2022). Adam: A Method for Stochastic Optimization. Retrieved 15 June 2022, from https://arxiv.org/abs/1412.6980

Kruglyak, I. (2021) Natural language processing: Tasks and application areas, Avenga. Available at: https://www.avenga.com/magazine/natural-language-processing-

application-areas/ (Accessed: January 16, 2023).

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2019). ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. Retrieved 13 June 2022, from https://arxiv.org/abs/1909.11942

Laskar, M., Huang, X., & Hoque, E. (2020). Contextualized Embeddings based Transformer Encoder for Sentence Similarity Modeling in Answer Selection Task. Retrieved 15 June 2022, from https://aclanthology.org/2020.

Leahy, S. (2019). Climate study warns of vanishing safety window—Here's why. (accessed July 1, 2019) https://www.nationalgeographic.com/environment/2019/03/climate-change-model-warns-of-difficult-future.

Lemarchand, P., McKeever, M., MacMahon, C., & Owende, P. (2022). A computational approach to evaluating curricular alignment to the united nations sustainable development goals. Frontiers in Sustainability, 74.

Lewis, M., Liu, Y., Goyal, N., Ghazvini Nejad, M., Mohamed, A., Levy, O., & Zettlemoyer, L. (2019). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv:1910.13461.

Liu, F., Shakeri, S., Yu, H., & Li, J. (2021). EncT5: Fine-tuning T5 Encoder for Non-autoregressive Tasks. Retrieved 15 June 2022, from https://arxiv.org/abs/2110.08426

Liu, P. J., Chung, Y. A., Ren, J. (2019). Summae: Zero-shot abstractive text summarization using length-agnostic auto-encoders. arXiv preprint arXiv:1910.00998.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., & Chen, D. et al. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. Retrieved 13 June 2022, from https://arxiv.org/abs/1907.11692

Martinez, D. (2021) Is transfer learning the final step for enabling AI in aviation? Available at: https://datascience.aero/transfer-learning-aviation/ (Accessed: January 16, 2023).

(MathWorks, 2021). Retrieved 28 August 2022, from https://www.mathworks.com /discovery/feature-extraction.html

Matsui, T., Suzuki, K. & Ando, K. (2022) A natural language processing model for supporting sustainable development goals: translating semantics, visualizing nexus, and connecting stakeholders. Sustain Sci 17, 969–985 (2022). https://doi.org/10.1007/s11625-022-01093-3

Merritt, R. (2022) What is a transformer model?, NVIDIA Blog. Available at: https://blogs.nvidia.com/blog/2022/03/25/what-is-a-transformer-model/ (Accessed: January 16, 2023).

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

Muralidhar, K.S.V. (2021) Learning curve to identify overfitting and underfitting in machine learning, Medium. Towards Data Science. (Accessed: January 31, 2023).

Nishant, R., Kennedy, M., & Corbett, J. (2020). Artificial intelligence for sustainability: Challenges, opportunities, and a research agenda. International Journal of Information Management, 53, 102104.

Ornes, S. (2022) Will transformers take over Artificial Intelligence?, Quanta Magazine. Available at: https://www.quantamagazine.org/will-transformers-take-over-artificial-intelligence-20220310/ (Accessed: January 16, 2023).

Papineni, Kishore Roukos, Salim Ward, Todd Zhu, Wei Jing. (2002). BLEU: a Method for Automatic Evaluation of Machine Translation. 10.3115/1073083.1073135.

Pramod (2020) How does NLP pre-processing actually work?, Medium. Available at: https://medium.com/predict/how-does-nlp-pre-processing-actually-work-8d097c179af1 (Accessed: January 16, 2023).

Pedersen, T., Patwardhan, S., & Michelizzi, J. (2004, July). WordNet:: Similarity-Measuring the Relatedness of Concepts. In AAAI (Vol. 4, pp. 25-29).

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).

Prechelt, L. (2012) Early stopping - but when?, SpringerLink. Springer Berlin Heidelberg. Available at: https://link.springer.com/chapter/10.1007/978-3-642-35289-8_5.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. OpenAI blog, 1(8), 9.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., & Matena, M. (2019). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. Retrieved 15 June 2022, from https://arxiv.org/abs/1910.10683

Reddy, S. (2019). GloVe and fastText — Two Popular Word Vector Models in NLP — SAP Blogs. Retrieved 28 August 2022, from https://blogs.sap.com/2019/07/03/glove-and-fasttext-two-popular-word-vector-models-in-nlp

Reza, S., Ferreira, M., Machado, J., & Tavares, J. (2022). A Multi-head Attention-based Transformer Model for Traffic Flow Forecasting with a Comparative Anal-

ysis to Recurrent Neural Networks. Expert Systems with Applications. 202. 1-11.10.1016/j.eswa.2022.117275.

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. ArXiv, abs/1910.01108.

Sung, C., Dhamecha, T., Saha, S., Ma, T., Reddy, V., & Arora, R. (2019). Pre-Training BERT on Domain Resources for Short Answer Grading. 6073-6077. 10.18653/v1/D19-1628.

Taylor, K. (2023) Trending NLP language model comprising of scope, HitechNectar. Available at: https://www.hitechnectar.com/blogs/here-are-the-top-nlp-language-models-that-you-need-to-know/ (Accessed: January 16, 2023).

Thomas W., Lysandre D., Victor S., Julien C., Clement D., Anthony M., Pierric C., Tim R., R'emi L., Morgan F., & Jamie B. 2019. Hugging Face's Transformers: State-of-the-art Natural Language Processing. ArXiv abs/1910.03771 (2019).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N. & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.

Vinuesa, R., Azizpour, H. & Leite, I. (2020) The role of artificial intelligence in achieving the Sustainable Development Goals. Nat Commun 11, 233 (2020). https://doi.org/10.1038/s41467-019-14108-y

Wang, W., Bi, B., Yan, M., Wu, C., Bao, Z., & Xia, J. et al. (2019). StructBERT: Incorporating Language Structures into Pre-training for Deep Language Understanding. Retrieved 13 June 2022, from https://arxiv.org/abs/1908.04577

Wilson (2023) Using natural language processing for supporting Sustainable De-

velopment Goals, Permutable.ai - Machine Learning Business Intelligence. Available at: https://permutable.ai/2022/09/06/using-natural-language-processing-for-supporting-sustainable-development-goals (Accessed: January 19, 2023).

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., & Le, Q. (2019). XLNet: Generalized Autoregressive Pretraining for Language Understanding. Retrieved 12 June 2022, from https://arxiv.org/abs/1906.08237

Zhang, X., Yu, F., Karaman, S., Zhang, W., & Chang, S. (2018). Heated-Up Softmax Embedding. Retrieved 15 June 2022, from https://openreview.net/forum?id=SkGpW3C5KX

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., ... & He, Q. (2020). A comprehensive survey on transfer learning. Proceedings of the IEEE, 109(1), 43-76.

Zhu, G., & Iglesias, C. (2017). Sematch: Semantic Similarity Framework for Knowledge Graphs. Knowledge-Based Systems. 130. 10.1016/j.knosys.2017.05.021.

8 Appendix

8.1 Appendix A: Description of Hyperparameters presented in Table 5

do_sample - The hyperparameter creates words based on their conditional probabilities. The library supports greedy decoding, a simple method that chooses the word that has the highest probability, or acting avariciously.

early_stopping_metric - In order to determine model over-fitting, the hyperparameter uses evaluation loss. Every epoch, the metric checks the evaluation loss to see if it is still changing or no longer changing significantly. Since the weights are highly optimised at this checkpoint, the metric then stops further training.

evaluation_batch_size - Sets batch size of evaluation set. Batch sizes defines the number of samples to work through before updating the weights of a model. Larger batch sizes require more memory and computational power whereas lower batch sizes require less.

evaluate_during_training - Tests the model on the evaluation set and calculates the evaluation loss.

evaluate_during_training_steps - The hyperparameter performs an evaluation after the number of steps that is specified. Every time an evaluation is performed, a checkpoint model and the evaluation outcome are saved.

evaluate_during_training_verbose - Prints results from evaluation during training as an output.

fp16 - Half precision floating point format or fp16 uses 16 bits for single preci-

sion. By using small batch sizes, this allows for the training of large models but is memory bandwidth sensitive.

learning_rate - Controls how quickly a model learns or updates the estimates of a parameter. It regulates the rate at which the model adjusts to the data.

max_sequence_length - Sets the longest sequence that the model will tolerate. The maximum length of sequences that can be passed to Transformer models is limited in different types of pre-trained models.

number_beams - Refers to beam search, used for text generation. It returns the n most probable next words, rather than greedy search.

number_training_epochs - Refers to the number of epochs the model trains for.

optimizer - Sets a default optimizer. Adaptive optimizers like AdamW and Stochastic Gradient Descenet (SGD) are default choices for training transformer models.

overwrite_output_directory - When a trained model is saved to the output directory, it will replace any previously saved models in the same location.

polynomial_decay_schedule_lr_end - Starts with a large learning rate and then decays it over time as training progresses which aids in better generalization of the model.

reprocess_input_data - Reprocessing is a good practice as it ensures appropriate formatting of the input data. Even if a cached file of the input data already exists in the directory, the input data will still be reprocessed.

save_evaluation_checkpoints - Every epoch ends with the saving of a model checkpoint. The best version of the model can be chosen from these saved checkpoints by visualising the evaluation loss.

save_steps - Saves a model checkpoint at every specified number of steps. Setting to -1 to disables this.

top_k & top_p - Two sampling techniques that determine range for picking output tokens.

training_batch_size - Set batch size of the training set.

use_early_stopping - Utilizes the early_stopping_metric to stop training when evaluation loss shows stability.

use_multiprocessing - Multiprocessing allows converting data into features. Enabling may speed up processing, but may cause instability in certain cases.

wandb_project - An experiment tracking tool for ML and AI. Sets the name of W&B project and logs all hyperparameter values, training losses, and evaluation losses to the given project. The project can be visualized for model training summary.

warmup_ratio - A mechanism for minimising the primacy effect of the initial training examples. Without it, it might take a few more epochs to reach the desired convergence. The ratio of total training steps where learning rate will warm up is determined by the hyperparameter.

warmup_steps - Number of training steps where learning rate will warm up. Overrides warmup_ratio.