Automated Analysis of Climate Policy Documents using Natural Language Processing

Master Thesis

by

Nicolas Becker

Degree Course: Industrial Engineering and Management M.Sc.

Matriculation Number: 2000106

Institute of Applied Informatics and Formal Description Methods (AIFB)

KIT Department of Economics and Management

Advisor:	Prof. Dr. Andreas Oberweis
Second Advisor:	Prof. Dr. J. Marius Zöllner
Supervisor:	M.Sc. Martin Forell
Submitted:	August 25, 2024

Abstract

Analysts face challenges in assessing climate policy documents due to their complexity, diverse content structures, and increasing volume. To address this, the present thesis explores the application of Natural Language Processing (NLP) and Large Language Models (LLMs) for automating the identification and classification of relevant information within these documents. A scalable end-to-end pipeline is proposed, leveraging general-purpose LLMs with prompting techniques to circumvent the need for fine-tuning. The pipeline, implemented as a prototype focusing on transport sector targets in Nationally Determined Contributions (NDCs), successfully automates the detection and classification of transport-related measures and targets within these documents. The results demonstrate the feasibility of using LLMs to assist policy analysts, offering a flexible and platform-agnostic framework that can be expanded to various languages and document types. This approach has the potential to significantly enhance the efficiency and accuracy of climate policy analysis.

Contents

List of Abbreviations

AI Artificial Intelligence

ANN Artificial Neural Network

BART Bidirectional and Auto-Regressive Transformer

BERT Bidirectional Encoder Representations from Transformer

BLEU BiLingual Evaluation Understudy

BMWK German Federal Ministry for Economic Affairs and Climate Action

BOW Bag-of-Words

CAP Climate Action Plan

CBOW Continuous Bag-of-Words

CDNK Climate and Development Knowledge Network

CFG Context-Free Grammar

CKY Cocke–Younger–Kasami

CL Computational Linguistics

CNN Convolutional Neural Network

CoT Chain-of-Thought

COP Conference of the Parties

DIA Document Image Analysis

ELMo Embeddings from Language Models

FA Factor Analysis

GIZ German Development Cooperation

GloVe Global Vectors

GPT Generative Pre-trained Transformer

IDOS German Institute of Development and Sustainability

IE Information Extraction

IKI International Climate Initiative

IPCC Intergovernmental Panel on Climate Change

IR Information Retrieval

KAI Knowledge Acquisition and Inferencing

KG Knowledge Graph

LLM Large Language Model

LM Language Model

LT-LEDS Long-Term Low Emission Development Strategy

LTS Long-Term Strategy

LSTM Long Short-Term Memory

ML Machine Learning

MLP Multilayer Perceptron

MNZ Mobilize Net-Zero

MT Machine Translation

NDC Nationally Determined Contribution

NECP National Energy and Climate Plan

NER Named Entity Recognition

NGO Non-governmental Institution

NLM Neural Language Model

NLG Natural Language Generation

NLP Natural Language Processing

NLU Natural Language Understanding

OCR Object Character Recognition

PCA Principal Component Analysis

PLM Pretrained Language Model

POS Part-of-Speech

QA Question Answering

RAG Retrieval-Augmented Generation

RL Reinforcement Learning

RNN Recurrent Neural Network

SDG Sustainable Development Goal

Seq2Seq Sequence-to-Sequence

SDSN United Nations Sustainable Development Solutions Network

SLM Statistical Language Model

SVM Support Vector Machine

TF-IDF Term Frequency–Inverse Document Frequency

TL Transfer Learning

t-SNE t-distributed Stochastic Neighbor Embedding

UNFCCC United Nations Framework Convention on Climate Change

List of Figures

List of Tables

1 Introduction

Governments face significant challenges in addressing anthropogenic climate change and its impacts. According to the 6th Synthesis Report by the Intergovernmental Panel on Climate Change (IPCC), published in March 2023, global surface temperature in the period between 2011 and 2020 has been registered to be on average 1.1°C higher than in the pre-industrial records from the period between 1850 and 1900 (Calvin et al., 2023). Coupled with a growing public awareness of the dangers of climate change and an increasing diversity of actors involved (Calvin et al., 2023), the pressure on policymakers to respond to public demands for targeted legislation and action plans increases. This is evidenced by the growing number of issued policy documents related to climate change (Żółkowski et al., 2022)(Szomszor & Adie, 2022).

Climate policy documents contain strategies and objectives envisioned by governments and international bodies to address the challenges related to climate change (Clark et al., 2023). Nationally Determined Contributions (NDCs) are considered of particular importance as they serve to communicate efforts towards this end to the international community. Within these documents, the Parties to the Paris Agreement (UNFCCC, 2016) set out their actions and targets to reduce greenhouse gas emissions and adapt to the consequences of climate change at the national level, therefore laying out their contributions towards the agreed upon objective of reducing global temperature increase to a maximum of 2°C in comparison to pre-industrial times.

In this context, NDCs are the subject of numerous analytical platforms that seek to present the content of the document, with the objective of enhancing the transparency and accessibility to a broader audience, either from a general perspective (Casas et al., 2021), or in relation to individual sectors, such as transport (International Transport Forum, 2018)). The required analytical process is conducted manually by policy analysts and climate experts. Due to their complex language and heterogeneous content structures, manual analysis of these documents represents a time-consuming and error-prone process. Additionally, the increasing number of publications and decentralized provision of documents present analysts with the challenge of producing up-to-date and holistic analyses. The process is resource-intensive, and in occasions relying on the assistance of volunteers, which limits the scalability and increases its organizational complexity (Juhasz et al., 2024).

Natural Language Processing (NLP) is a research branch of Artificial Intelligence (AI) that has recently gained attention since the launch of publicly available chat applications, the most famous being OpenAI's ChatGPT (W. X. Zhao et al., 2023). NLP is proving useful in various contexts for assistance with the analysis of significant amounts of textual data (Arowosegbe & Oyelade, 2023)(Lareyre et al., 2023)(Tounsi & Temimi, 2023)(Tyagi

& Bhushan, 2023). In particular, NLP techniques promise to reduce the effort and bias associated with the manual extraction of information from written or spoken text.

Within NLP, Machine Learning (ML) methods are applied to develop Large Language Models (LLMs) with the capability to understand and/or generate natural language. These models are trained on extensive amounts of text data to acquire a profound comprehension of language patterns, semantics, and context (W. X. Zhao et al., 2023). The ongoing development of increasingly powerful models is accompanied by an emphasis on the advancement of prompting technologies that facilitate the effective utilization of LLMs (Saravia, 2022). These developments have led to a progressive expansion in the potential applications of automation with NLP.

1.1 Objective

The present thesis intends to explore whether LLMs are potentially applicable in the context of climate policy, offering automated assistance to policy analysts in order to gain insights from extensive amounts of climate policy documents. To this end, this work proposes a scalable end-to-end pipeline that applies LLMs to the tasks of identifying, retrieving, processing and classifying information from climate policy documents. The pipeline is designed to provide suggestions for relevant information contained within the document at hand, reducing time, efforts, and the susceptibility to human error entailed with the manual process.

Specifically, the selected approach is to utilize general-purpose LLMs, through prompting techniques to elicit the desired output for the specified tasks. This notion is the foundation in the following research hypothesis:

> *State-of-the-art LLMs are capable of performing the desired task of extracting relevant information from climate policy documents and perform corresponding classification on the extracted text, circumventing the need for additional training or fine-tuning of the models on climate domain data. Thus, through proper application of LLMs, it is possible to automize the task of climate policy analysis.*

In order to prove this hypothesis, a pipeline concept is introduced, specifying the constituting modules and components. The proposed concept is implemented as a prototype in the context of a specific use case related to the analysis of United Nations Framework Convention on Climate Change (UNFCCC) submissions from a transport sector perspective. In this context, particular focus is set on NDCs.

Although the implemented prototype is designed with the specific intention of analyzing

UNFCCC submissions, the proposed concept is intended to be generalizable to a variety of documents in varying formats and languages.

1.2 Structure

The structure of the remaining chapters of this thesis is outlined as follows. Chapter 2 provides an introduction to the fundamental concepts associated with ML, NLP and climate policy. Chapter 3 continues by delineating the related work associated to the particular intersection between NLP and climate research, or more specifically, climate policy. Subsequently, the elaborated concept is introduced in Chapter 4 with prior definition of the associated challenges and requirements to the task of automated climate policy analysis. In order to evaluate the performance of the designed concept, a prototype is implemented for a specific use case, which is described in Chapter 5. The performance evaluation of the prototype is conducted in Chapter 6, followed by a discussion of the obtained results in Chapter 7. The discussion is guided by the formulated requirements, which are revisited at this point. Finally, Chapter 8 concludes this thesis with a concluding remark and an outlook on possible future work in this topic.

2 Foundations

The methodological foundation of this thesis is formed by two research areas, namely, ML and NLP. The following chapter provides an introduction to these two areas. Chapter 2.1 addresses the topic of ML, while Chapter 2.2 offers an insight into research pertaining to NLP, with a particular focus on language models. Finally, Chapter 2.3 introduces the climate policy domain, which represents the thematic focus of this work.

2.1 Machine Learning

While AI, defined as the ability of a machine to display capabilities that are generally attributed to humans, such as reasoning, learning, planning, and creativity (European Parliament, 2020), has recently been reshaping the way to approach tasks and face challenges (Maslej et al., 2024), the main theoretical foundations in the field date back to the 1940s and 1950s (Russell et al., 2022), when the first attempts to mimic the human brain with mathematical operations were introduced (McCulloch & Pitts, 1943). Modern state-of-the-art models draw upon these findings (Goodfellow et al., 2016)(Russell et al., 2022).

In particular, the ability of a computer system to *learn*, encompassing the acquisition, retention, and application of knowledge, is of interest when creating an AI system. The idea of enabling machines to learn without explicit programming (Samuel, 1959), serves as a foundational principle for the research area of ML, a subfield to AI. The learning process is based on data, that is facilitated by humans and algorithms. The machine leverages the data in order to utilize the information that it contains.

In the context of this thesis, the concept of *data* provided by ISO/IEC 25012:2008 (2008) is applied, which defines data as the representation of information in a formal manner. Meanwhile, *information* is defined as knowledge related to objects, such as facts, ideas, or concepts, that acquire a specific meaning in a given context (Batini & Scannapieco, 2006).

The research field of ML is typically divided into three distinct subcategories according to the paradigms applied (Murphy, 2022)(Russell et al., 2022), as illustrated in Figure 1. These distinctions are delineated in the following sections.

2.1.1 Supervised Learning

The most prevalent machine learning methodology is supervised learning, which entails the training of machines on data from experience collected in the past with the objective of making predictions or decisions on future data (Murphy, 2022). The data is contained

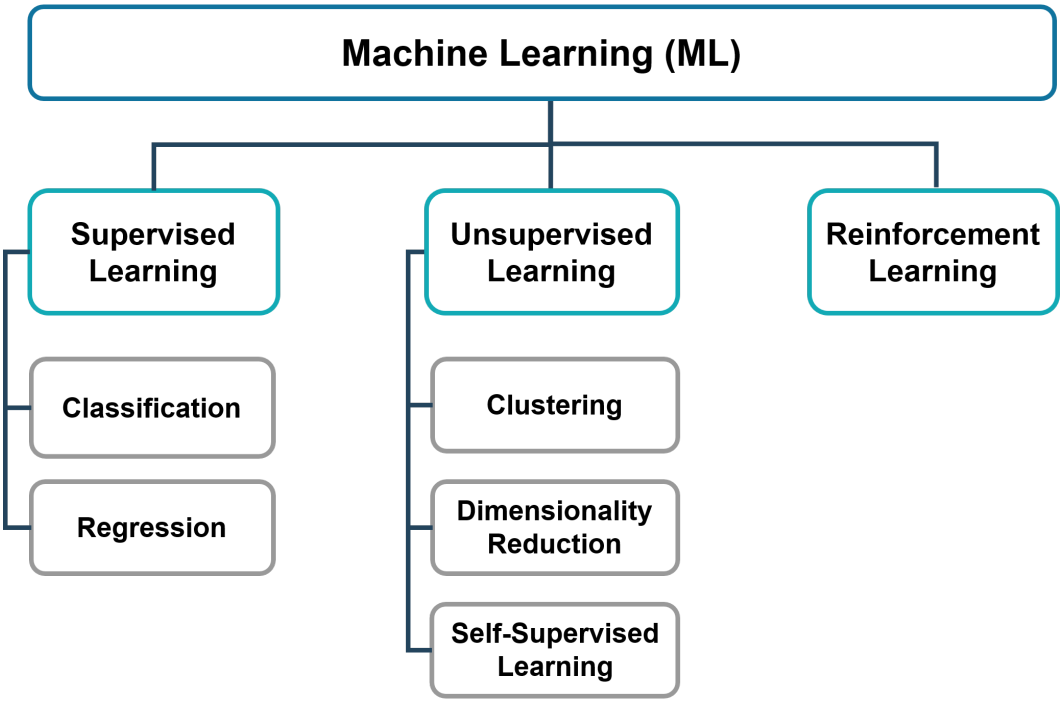


Figure 1: Categorization of ML paradigms, according to Murphy (2022).

in a training set $D = \{(x_n, y_n) : n = 1, ..., N\}$, which consists of observations $x \in X$ from a predefined feature space X and corresponding output representations $y \in Y$, also called targets or labels. The machine is trained to learn a mapping function $f : X \rightarrow Y$ from the input space to the output space, which best fits both the previously seen data and future unseen data, given a respective performance measure. The learned mapping function $f : X \rightarrow Y$ is also referred to as model. Depending on the characteristics of the output space, supervised learning tasks can be distinguished between classification tasks and regression tasks (Murphy, 2022).

Classification

Typically, supervised methods are applied to classify data with unknown class membership (Han et al., 2012). In this case, the output space comprises a finite set of C mutually exclusive class labels, $Y = (y_1, y_2, \ldots, y_C)$. The training data set provides labelled sample data, i.e., data with known class membership. Subsequently, learning algorithms are employed to identify patterns in the data that can be generalized to examples outside the training sample. The known labels serve a reference point for supervision during the learning process, as they enable the machine to identify misclassifications and adapt the learned classifier model accordingly. The validation of the classifiers is conducted using test data, which is a subset of the labelled data set, whose class variables are known but were withheld from the classifier during the training process (Han et al., 2012).

Traditional ML methodologies apply a variety of distinct classifier models, including Decision Trees, Random Forests, Support Vector Machines (SVMs) and Neural Networks (Han et al., 2012). The latter employ analogies to the human brain by representing a structure of artificial neurons and weighted connections between them, therefore they are referred to as Artificial Neural Networks (ANNs) (Murphy, 2022). The original version of ANNs is described in the concept of feedforward neural networks or Multilayer Perceptrons (MLPs), which describe a mathematical function mapping input values to output values. An input layer receives the information from the input data set and forwards it through a number of hidden layers to an output layer, from which a classification result emerges. The weighting between two nodes, which represents the strength of connection between two adjacent neurons, is adjusted during model training. In order to achieve this, the predicted labels are compared with the known labels from the training data set, and the classification error is determined as the deviation between the two values. The error is then passed backwards through the neural network, whereby the adjustments are made to the model's parameters in order to reduce the output error. This process is named *backpropagation* (Murphy, 2022). The output of the the final hidden layer of the network is typically normalized with the softmax function to obtain a probability distribution over the K predicted output classes:

$$softmax(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}, \quad for\ i = 1, 2, \ldots, K. \tag{2.1}$$

It can be observed that ANNs achieve good generalizability compared to the previous models. However, the abstract knowledge representation inherent to ANNs leads to reduced explainability and these models require large amounts of data during training. With increasing data availability and computational power, the size and complexity of ANNs has augmented in the past years, giving rise to a new research field referred to as Deep Learning (Fleuret, 2023)(Goodfellow et al., 2016)(Russell et al., 2022). In this context, "depth" refers to the extensive number of hidden layers present in deep neural networks. The increasing number of layers enables these models to process complex and high-dimensional input representations. This ability has especially led to advances in the field of image processing, where the surging of Convolutional Neural Networks (CNNs) has set new standards on the ImageNet competition in 2012 (Krizhevsky et al., 2017), setting the stage for an intensification of research in deep learning applications in a variety of fields.

Regression

Regression is applied in cases when the target variable is continuous, as opposed to categorical classification targets. Instead of predicting class membership, regression models are designed to predict a real-valued output $y \in Y$ based on input features $x \in X$ present in the training data (Murphy, 2022). The output space in regression is typically unbounded, allowing for a wide range of possible values, e.g. $Y = \mathbb{R}$. Linear regression assumes a linear dependence of the target variable y to the input features x. It therefore models their relationship as a linear function. Mathematically, this can be represented as $\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_m x_m + \varepsilon$, where $\hat{y}$ is the predicted output, $x_1, x_2, \ldots, x_m$ are the input features, $\beta_1, \beta_2, \ldots, \beta_m$ are the model coefficients to be learned, and ε represents the error term (Murphy, 2022).

2.1.2 Unsupervised Learning

Learning paradigms that do not rely on the provision of labelled training data fall under the category of unsupervised learning. In contrast to supervised learning, the objective of unsupervised learning is to identify patterns, structures, and intrinsic relationships within the data, rather than to predict outputs. Consequently, the provided data is limited to inputs $D = \{x_n : n = 1, ..., N\}$, without the corresponding outputs y_n. This circumvents the data labelling task, which is frequently performed manually in a time-consuming and expensive process (Murphy, 2022). The two most frequent tasks within unsupervised learning are clustering and dimensionality reduction, while self-supervised has just recently evolved as a third popular paradigm.

Clustering

Clustering algorithms partition the input data into subsets (*clusters*) containing similar data points based on their intrinsic characteristics. Consequently, data points contained within a cluster exhibit a high degree of similarity, while points from different clusters exhibit a low degree of similarity (Han et al., 2012). Subsequently, the resulting groups can be analyzed to identify the relevant features that lead to the division of the data points, revealing associations that may be invisible on first sight. Examples including K-means clustering and hierarchical clustering (Han et al., 2012). Figure 2 shows a distribution of data points corresponding to word representations in two-dimensional space. The application of clustering yields a clear separation of policy related words (*bottom-right*) from the remaining climate-related set of words. Within these, further separations is possible for the emissions cluster (*top-right*), the environmental cluster (*bottom-center*) and the technology-related cluster (*top-left*).

Dimensionality Reduction

Dimensionality reduction techniques aim to reduce the complexity of data by transforming it into a lower-dimensional space, while preserving its essential information (Murphy, 2022). The lower-dimensional representation of a data set $x_n \in \mathbb{R}^D$ are also referred to as latent factors $z_n \in \mathbb{R}^K$, where $K < D$. These latent factors are assumed to be generating the observed data, making latent Factor Analysis (FA) a similar problem to the

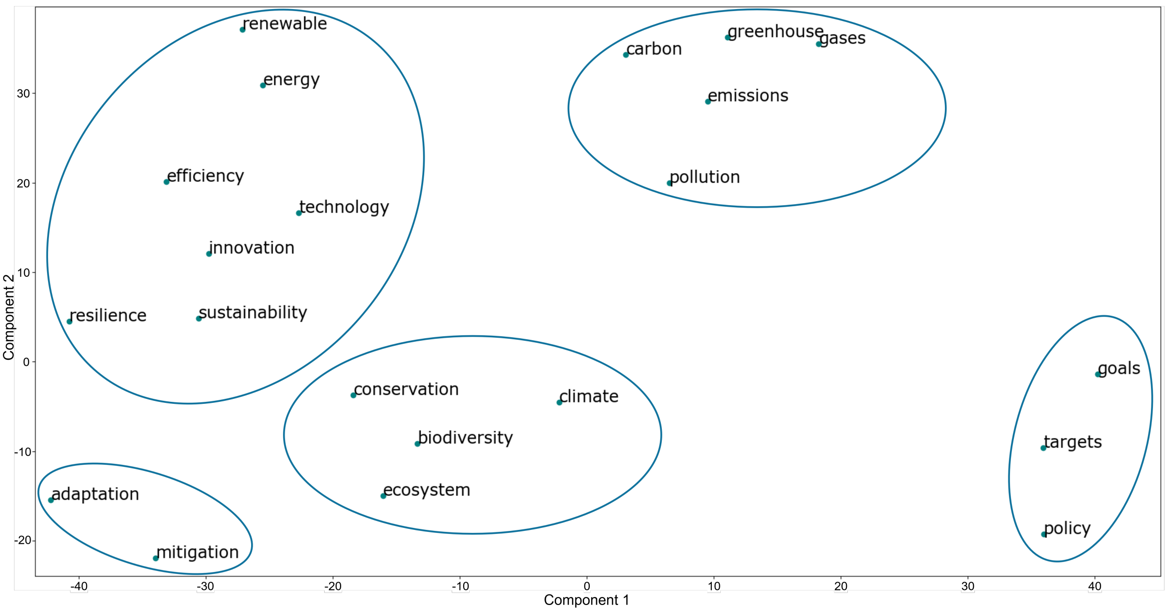


Figure 2: Visualization of word meanings and formation of clusters in two-dimensional space through application of t-SNE. Own illustration based on Li et al. (2016).

regression task. Principal Component Analysis (PCA) is a special case of factor analysis and the most commonly used method for dimensionality reduction. It plays a crucial role in exploratory data analysis, data preprocessing, and feature extraction (Murphy, 2022), since smaller-dimensional vectors are easier to process for both humans and machines. A method which is typically used for visualization of high-dimensional vectors into two or three-dimensional space is t-distributed Stochastic Neighbor Embedding (t-SNE) (van der Maaten & Hinton, 2008). An example for the application of t-SNE is provided in Figure 2, in which high-dimensional vectors from representing words are projected in two-dimensional space. This uncovers the relations between individual data points, visualizing the effect of modifications and negations on proximity in underlying word meaning. Particularly in the context of proximity, dimension reduction plays a key role, as high-dimensional data is subject to a phenomenon known as the *curse of dimensionality*, which refers to increasing sparsity of data in high-dimensional spaces. Sparsity has negative effects on the calculation of distance metrics (Altman & Krzywinski, 2018).

Self-Supervised Learning

Approaches which entail the creation of proxy labeled training examples $(x_1, \hat{x}_2)$ from unlabeled data are referred to as self-supervised learning (Murphy, 2022). The idea behind this approach is that the resulting supervised learning task of inferring the proxy output $\hat{x}_2$ from the input x_1 enables the application of established supervised learning techniques to learn a model $f : X \setminus \{x_2\} \rightarrow X$, thus circumventing the challenging task of inferring the latent factors z that underlie the observed data, in order to learn useful features from the data (Murphy, 2022).

2.1.3 Reinforcement Learning

The third main learning paradigm in ML is Reinforcement Learning (RL). The core idea is that the machine, in this case referred to as an agent, learns to make decisions by interacting with an environment it is placed at (Mitchell, 2013). The task environment is typically fully specified by a PEAS description, which refers to Performance, Environment, Actuators, and Sensors (Russell et al., 2022). Similar to unsupervised learning, reinforcement learning does not rely on labelled data, but collects experience from its interactions with the specified environment. Hence, no previously collected data set is required. The agent performs actions in the environment and receives feedback from a critic or reward function in the form of rewards or penalties. These indicate the quality of the agent's decisions based on the desirability of the outcome. The goal of reinforcement learning is to learn a policy π, which is a mapping $\pi : X \rightarrow A$ from an input $x \in X$ derived from environment states to a suggested action $a \in A$, that maximizes the cumulative reward over time (Murphy, 2022). The agent iteratively explores the environment and utilizes the gained experience in order to identify the most effective policy, which results in the highest reward or lowest penalty. This process is specified in common algorithms named value iteration and policy iteration (Sutton & Barto, 2018) that are applied in traditional RL techniques such as Q-Learning (Watkins, 1989). Reinforcement learning has been applied in a multitude of domains, with the most famous applications being those related to game playing, ranging from Samuel's checkers player (Samuel, 1959) to AlphaGo (Silver et al., 2016).

2.1.4 Evaluation Metrics

It is possible to evaluate the performance of ML models on the basis of various aspects, including runtime, scalability, or robustness to outliers in the training data (Han et al., 2012). In the context of classification tasks, which are of particular relevance to this thesis, models are typically evaluated based on their ability to correctly predict data from a previously unseen test data set. The *accuracy* measure describes the proportion of tuples in the test data set that were correctly classified. In the binary case with two classes (positive P and negative N), this is expressed with the formula

$$Accuracy = \frac{(TP + TN)}{(TP + FP + TN + FN)}, \tag{2.2}$$

consisting of four components:

- *True-positives* (TP) refer to positive tuples that have been correctly classified as such.
- *True-negatives* (TN) denote negative tuples that have been correctly classified as such.
- *False-positives* (FP) refer to instances where positive tuples have been incorrectly classified as negative.
- *False-negatives* (TN) represent negative tuples that have been incorrectly classified as positive.

These four components of the classification result are generally recorded in a *confusion matrix*, which is also possible to extend for the case of multi-label classification with more than two classes (Han et al., 2012).

Two additional measures are commonly employed in the scientific literature, which are themselves derived from the listed components. The *precision* denotes the proportion of tuples that have been correctly classified as positive (TP) in relation to the total number of tuples:

$$Precision = \frac{TP}{(TP + FP)}. \tag{2.3}$$

In contrast, *recall* is applied to describe the proportion of positive tuples in the test data set that have been correctly classified as such:

$$Recall = \frac{TP}{(TP + FN)}. \tag{2.4}$$

A trade-off effect is observed between these two measures, meaning that an improvement in one can be achieved at the expense of a deterioration in the other (Han et al., 2012). In order to balance this trade-off, both measures are frequently combined to form a third measure, the F_β*-score*:

$$F_\beta = \frac{(1 + \beta^2) * Precision * Recall}{\beta^2 * Precision + Recall}. \tag{2.5}$$

By selection of β, it is possible to determine which of the two measures should be given greater weight. The recall is given β-times the weight assigned to precision. The F_1-*score* (or *F-score*) is parametrized with $\beta = 1$, which assigns the same weighting to both measures.

2.2 Natural Language Processing

Language can be defined as a system for communication, comprising a set of symbols which are combined according to a set of rules and used for conveying information (Khurana et al., 2023). Following this definition, natural language is a system of communication that has evolved naturally in humans, without conscious planning (Mandl, 2019). This encompasses spoken, signed, or written forms of symbolic communication.

NLP is a discipline of ML that deals with the question of how a machine can understand and apply human natural language. In the same way language enables human individuals to communicate with one another and exchange information, language plays a key role in the interaction between humans and machines. With this in mind, NLP is to be seen as a technology that creates an interface between humans and machines, providing intuitive access to AI systems.

Russell et al. (2022) identify three reasons why it is of scientific interest for machines to learn human language:

1. NLP naturalizes communication between humans and machines, providing additional layers of interaction and, thus, increasing accessibility and convenience.
2. The ability to understand natural language provides machines with access to a vast repository of knowledge, which is available in natural language in sources such as encyclopedias or the internet.
3. A deeper understanding of natural language itself and its application is gained in the course of research in the field of NLP.

The research field of NLP is generally divided into two subfields. Natural Language Understanding (NLU) refers to techniques and methods by which machines gain an understanding of the meaning of spoken and written human language. NLU covers the analysis of syntax (structural relations), semantics (meaning) and pragmatics (context) in linguistic structures (Lee, 2024). This field of research is thus closely related to linguistics, in particular to the field of Computational Linguistics (CL), which is dedicated to the examination of natural language properties through the application of computational methods (Rao & McMahan, 2019). The field of NLU is divided into a number of subtasks and disciplines, such as Text Classification, Named Entity Recognition (NER), Part-of-Speech (POS) tagging, Topic Modeling and text-to-speech synthesis (or the inverse process).

Natural Language Generation (NLG) represents the second component of NLP (Lee, 2024). In this context, the machine is trained to generate language that is both coherent and comprehensible to humans, based on an internal representation of information

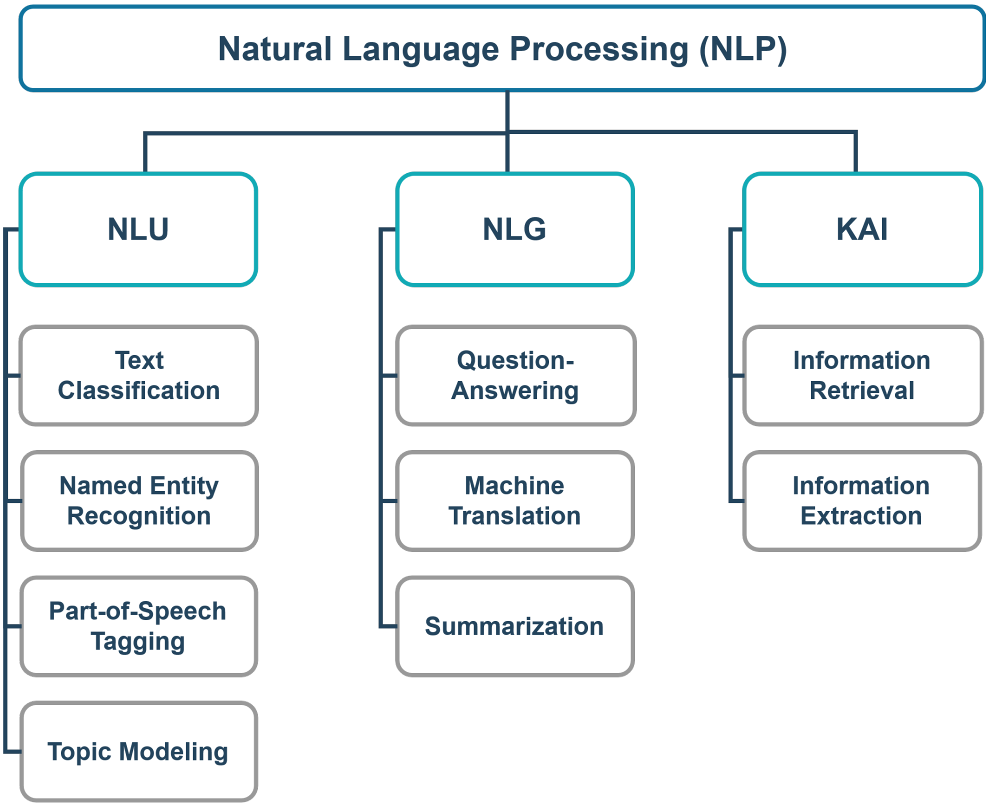


Figure 3: Categorization of NLP subfields, according to Lee (2024), and additional assignment of common NLP tasks.

(Khurana et al., 2023). The internal representation of information is derived from a previous encoding of natural language. Consequently, this component of NLP is commonly referred to as decoding, particularly in the context of deep learning-based methods. Common NLP tasks that include a NLG component are Question Answering (QA), Machine Translation (MT) or Summarization.

In certain use cases, where external knowledge is retrieved to generate the output, Knowledge Acquisition and Inferencing (KAI) can be included as a third component of NLP. External knowledge is required when relevant information for the output generation is not present in the internal representation of the input and context. It is therefore an intermediate step between NLU and NLG, which may be included after the linguistic input is fully processed (Lee, 2024). Typical tasks involved are Information Retrieval (IR) and Information Extraction (IE). Figure 3 illustrates the described categorization on NLP.

The following section will first describe, how text can be represented in a form that is understandable to a machine. Subsequently, it will address the topic of language modeling, which is a constituent element of NLP. The chapter concludes with introductions to the topics of Prompt Engineering and IR, which are of special importance to this thesis.

2.2.1 Computational Linguistics and Word Representations

At the core of language lies a hierarchy of fundamental elements that enable communication: *words*, *collocations* and *sentences*. Words serve as the building blocks of language, representing individual concepts, objects, actions, or ideas. Two or more words form collocations, which are also referred to as expressions. These collocations correspond to conventional ways of expressing information, including phrasal verbs (*phase out*) and noun phrases (*fossil fuels*) for the English language (Manning & Schütze, 1999). Sentences, on the other hand, provide structure and coherence to language by organizing words into closed and coherent sequences that convey thoughts or messages. Sentences can vary in complexity, from simple declarative statements to intricate compositions (Jurafsky & Martin, 2023). The focus of the present work is on written language. Consequently, a collection of sentences will be denoted as document D, while the entirety of words w contained in the ground population of text under analysis is called corpus T of text, composed of words from a vocabulary V.

A *grammar* is defined as a set of rules governing the arrangement and combination of words in a language, providing the framework for constructing grammatically correct sentences. The process of analyzing a sequence of words for adherence to a defined grammar is called parsing (Russell et al., 2022). The analysis encompasses syntax, which dictates the order and structure of words within sentences. Special importance in syntactic parsing lies within Context-Free Grammars (CFGs), which constitute the foundation of numerous formal models of natural language syntax (Jurafsky & Martin, 2023). Parsers such as the Cocke–Younger–Kasami (CKY) algorithm (Kasami, 1965) are able to decide whether a certain phrase or sentence adheres to a context-free grammar. This kind of technology is used in applications such as grammar checking and constituted an important role in early approaches toward language understanding. However, in modern NLU models, syntactic parsing is only implicitly performed within neural networks.

The semantic layer of language is concerned with the meaning of words and the textual structures of which they are a part. In order for a machine to comprehend the meaning of text, it is necessary to transform the text into a format that is understandable to a machine. This entails converting the input strings into a numerical representation, referred to as *word representations*. One approach to achieving this is through *one-hot encoding*, which involves constructing a vector for each word with the dimension corresponding to the size of the underlying vocabulary. Each word is represented by a single dimension of the vector, which is encoded in a binary format. The activation of a single dimension indicates which word is represented by the vector (Lee, 2024). To illustrate, the sentence *"renewable energy reduces carbon emissions."* is represented by the vectors:

- $[1, 0, 0, 0, 0]$ (renewable)

- $[0,1,0,0,0]$ (energy)
- $[0,0,1,0,0]$ (reduces)
- $[0,0,0,1,0]$ (carbon)
- $[0,0,0,0,1]$ (emissions)

A significant challenge associated with this representation is the absence of semantic information encoded within the vectors. The orthogonal arrangement of the vectors implies that the words “renewable” and “energy” will exhibit the same proximity as “energy” and “reduces”, not allowing the possibility of discerning any inherent semantic relationship between them. The attribution of one dimension to each word in the corpus can also rapidly result in large, high-dimensional vectors that are only sparsely filled. With regard to distance calculations, increased sparsity in higher dimensions leads to greater distances among vectors, a phenomenon described as *curse of dimensionality* (see Section 2.1.2). Given that similarity in word meaning is represented by proximity of word vectors, a main requirement of semantic word representation is not met by this approach.

The interpretation of a word w_t's meaning is heavily influenced by the word's context, which encompasses the set of k words preceding and the k words succeeding it, as denoted in Eq. 2.6. This notion is encapsulated in the observation of the English linguist J.R. Firth (1957): *“you shall know a word by the company it keeps”*. Firth's observation is referred to as the distributional hypothesis, which describes a link between the distribution of words in a given context and the meaning of a word (Jurafsky & Martin, 2023). This hypothesis is applied when building *word embeddings*, defined as a learned vector representation of a word or a word sequence based on distributions in the surrounding text (Jurafsky & Martin, 2023). With dimension sizes starting in the double-digit range (Nussbaum et al., 2024), these representations provide comparatively dense vectors, capable of storing enough information for semantic analysis. The value range of these dense representations is continuous and includes negative values.

$$\dots \; w_{t-(k+1)} \; \boxed{w_{t-k} \; w_{t-(k-1)} \; \dots \; w_{t-1}} \; w_t \; \boxed{w_{t+1} \; \dots \; w_{t+(k-1)} \; w_{t+k}} \; w_{t+(k+1)} \; \dots \tag{2.6}$$

Word2Vec (Mikolov et al., 2013) is an example of an approach to learn text embeddings, where the embeddings are learned as a by-product in the process of training a shallow neural network with a single hidden layer. Two different learning tasks are proposed by Mikolov et al. (2013). In the Continuous Bag-of-Words (CBOW) model, the classifier learns to predict a center word given the surrounding context words. Conversely, the Continuous Skip-gram model focuses on predicting context words given a single center word, emphasizing the relationships between a target word and its neighbors. The difference between both approaches is depicted in Figure 4. As a result, the captured semantic

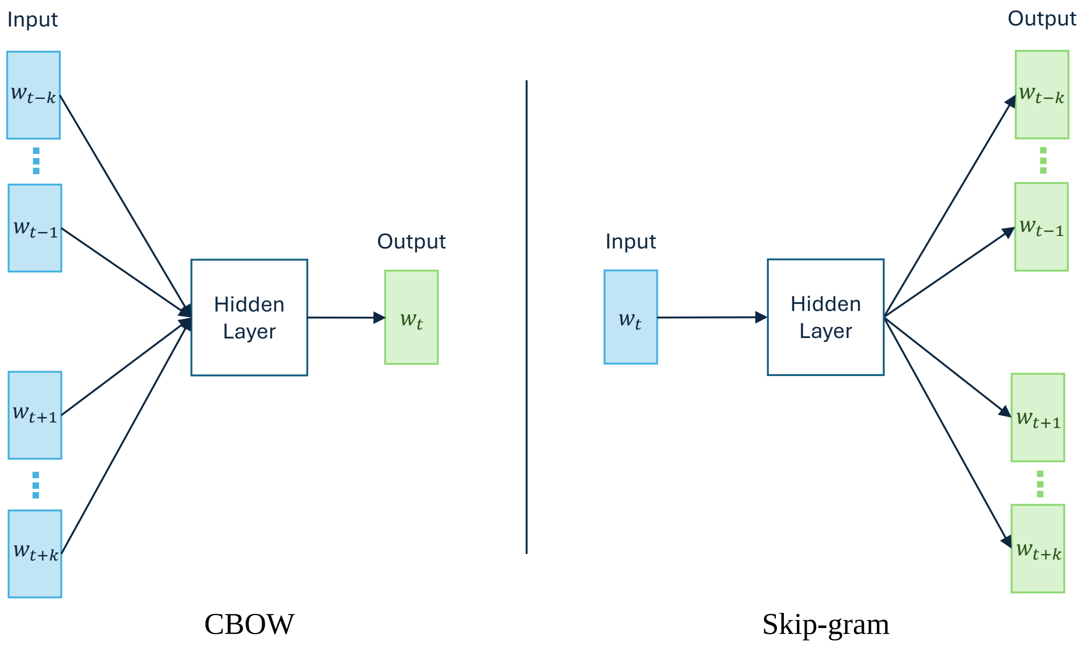


Figure 4: Illustration of the two proposed Word2Vec learning tasks. The CBOW model *left* predicts a target word w_t given the words in the $2k$ context window $\{w_{t-k},\ldots,w_{t-1}\}\bigcup\{w_{t+1},\ldots,w_{t+k}\}$, and the Skip-gram model *right* predicts context words in the window given the target word. Adapted from Mikolov et al. (2013).

information in the vectors is shown by examples of linear operations between word representation, such as *Paris - France + Japan = Tokyo* (Mikolov et al., 2013). Faster training and better scalability on larger corpuses is reached by a different approach called Global Vectors (GloVe) (Pennington et al., 2014). In this case, the embeddings are learned in an unsupervised manner by counting word co-occurrences in the corpus. Figure 2 shows word embeddings created with GloVe.

Embeddings are also effective in addressing one of the major challenges in NLU, namely polysemy. This phenomenon of natural language, where one word may bear different meanings in separate contexts, is addressed by superposition of individual word vectors of each sense (Arora et al., 2018). This is possible due to the sufficient sparsity of vectors to be split into corresponding components for each sense. The challenge of polysemy and in particular word sense disambiguation over time is further approached with the introduction of *dynamic* embeddings (Rudolph & Blei, 2018). As opposed to previous *static* models, these models include contextual information about time and social space, allowing for the continuous capture of semantic drift and variations in word meanings over time (Hofmann et al., 2021). The introduction of dynamic contextualized word embeddings such as Embeddings from Language Models (ELMo) (Peters et al., 2018) led to improvements in a variety of NLP tasks.

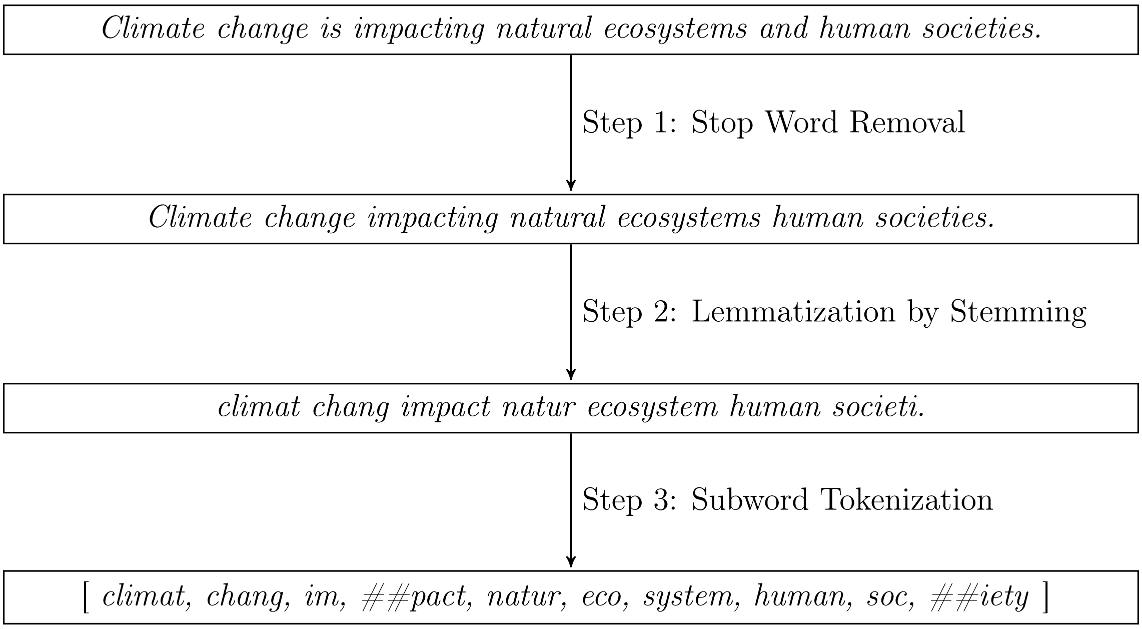

Climate change is impacting natural ecosystems and human societies.
Step 1: Stop Word Removal
Climate change impacting natural ecosystems human societies.
Step 2: Lemmatization by Stemming
climat chang impact natur ecosystem human societi.
Step 3: Subword Tokenization
[climat, chang, im, ##pact, natur, eco, system, human, soc, ##iety]

Figure 5: Illustrative text normalization procedure for the input sentence *"Climate change is impacting natural ecosystems and human societies"*..

The effectiveness of embeddings and word representations can be enhanced by prior normalization of the input text. Common techniques include *lemmatization* (which refers to the reduction of words to their roots, e.g., by deleting suffixes), the removal of *stop words* (frequent words which convey a limited amount of semantic information), and the reduction of text to tokens, a process named *tokenization*. The latter is specifically designed to increase the generalization capability of word models, thereby enabling the machine to comprehend words that were previously unseen in the training corpus. *Subword* tokenization decomposes words into modules, providing hints about possible relationships to other words present in the corpus with matching subwords (Jurafsky & Martin, 2023). An example for the entire normalization procedure is provided in Figure 5.

Once generated, word embeddings can be applied using other models, such as Random Forests, to enhance a variety of NLP tasks (H. Chen et al., 2022)(Ge & Moh, 2017). Common applications include search, by ranking results by similarity to an embedded query string; clustering, by grouping embedded text strings by similarity; and classification tasks, by classifying embedded text strings by their most similar label.

A comprehensive overview and categorization of approaches applied for the creation of word embeddings is provided by Torregrossa et al. (2021).

2.2.2 Language Models

The task of NLG is closely related to the concept of language modeling, which is the task of predicting a word, given a word's context (Russell et al., 2022). A model that is able to perform this task is referred to as Language Model (LM). Given a word sequence, a LM computes a probability distribution over possible subsequent words and draws from this probability distribution to generate a completion. The task of NLG can therefore be summarized as the process of repeated sampling over a word distribution. By computing the product of probabilities for each individual word in a sentence or sequence of words, a LM is capable of assigning a probability to a piece of text, thus correcting grammar or spelling errors and therefore assuming the role of a syntactic or semantic parser (Jurafsky & Martin, 2023).

The development of LMs can be grouped into four stages, namely Statistical Language Models (SLMs), Neural Language Models (NLMs), Pretrained Language Models (PLMs) and LLMs (W. X. Zhao et al., 2023), which are further discussed in the following.

Statistical Language Models

SLMs are early approaches to language modeling that apply statistical learning methods to learn the probability of word occurrence from a given text corpus. The application of naive Bayes on a corpus of text, assigning a probability to a word given the category or topic of the underlying text, is referred to as Bag-of-Words (BOW) method (Russell et al., 2022). This approach is limited by the absence of syntactical information and the assumption of independence between words. Consequently, the BOW method constitutes a simplified language model that fails to yield a coherent text, but does allow performing classification on a given text.

An extension of the BOW approach, which computes probabilities not for individual words but for word concatenations or sequences, is the n-gram model. The n-gram model regards sentences as a Markov chain in which each word to be generated depends on the $(n-1)$ previous words, a simplification denoted as Markov assumption (Jurafsky & Martin, 2023). Though the increase of the context window through an incrementing parameter n leads to improved generation capabilities, this is accompanied by the curse of dimensionality, which leads to sparsity and storage problems. The first refers to a possible absence of $(n-1)$-predecessors of a word in the corpus, and the latter to the exponentially increasing number of transitions to be estimated with increasing context size (W. X. Zhao et al., 2023). Further limitations of the models as n increases include a tendency to reproduce large passages from the training data verbatim (Russell et al., 2022). Despite it limitations, n-grams are fastly learnt models that perform well for the tasks of text classification and NER.

Neural Language Models

A more complex stage of language models is defined as NLMs (W. X. Zhao et al., 2023). They leverage neural networks to assess word probability in word sequences and therefore to build probabilistic classifier models. The Neural Probabilistic Language Model introduced by Bengio et al. (2003) addresses the curse of dimensionality suffered by previous n-gram-approaches through embedded representations learned by the model, allowing it to provide predictions over previously unseen word sequences through vector similarity. The limitations of this approach include the fixed context size defined though a sliding window over the input text and the large influence of word order in the input to the net.

The evaluation of varying context sizes was made possible through the introduction of recurrent neural structures to the field of NLP by Mikolov et al. (2010). Recurrent Neural Networks (RNNs) reuse input weights on an arbitrary amount of input variables while conserving the current state of input analysis in a separate parameter named hidden state h_t, with t referring to the respective time step of input computation. This allows the model to remain the same size when the context size increases, while keeping a memory of previously seen input. The model parameters are trained via backpropagation through time (Werbos, 1988), where the gradients are propagated through the length of context in the training data. This approach is affected by the vanishing gradients problem (Bengio et al., 1994), as the influence of further away input decreases rapidly, limiting the model's input processing capacity to a short-term memory of only a few time steps. Long Short-Term Memorys (LSTMs) (Hochreiter & Schmidhuber, 1997) approach this problem by extending RNNs with a long-term memory contained in a cell state c_t. The model contains additional weights that operate sequentially on the current input and hidden state, and define which information is no longer needed and forgotten, and which information is probably needed in future time steps and captured in the cell state.

A flagship task of NLP that has specifically benefited from the development of NLMs is Machine Translation (MT), which refers to the automatic translation of text from one language to another (Khurana et al., 2023). This task is particularly challenging, as the generated text in the destination language, besides being fluent, must fulfil the additional condition of adequacy or accuracy, by correctly translating diverging meanings of expressions across languages and taking into account variations in sentence length or word order (Lee, 2024). This further requirement provides models performing this task the designation of Conditional Language Models (Jurafsky & Martin, 2023). Similar tasks which generate output conditional on the input are question answering (QA) and summarization.

To perform the task of MT, the neural architecture of encoder-decoder networks or Seq2Seq models has been introduced. As illustrated in Figure 6, they contain two RNNs, one performing the task of encoding the input text in the language of origin (*encoder*) and the second generating corresponding text in the language of destination through decoding of the encoded representation of information (*decoder*). The final hidden state of the encoder network is therefore the input to the decoder network. In general terms,

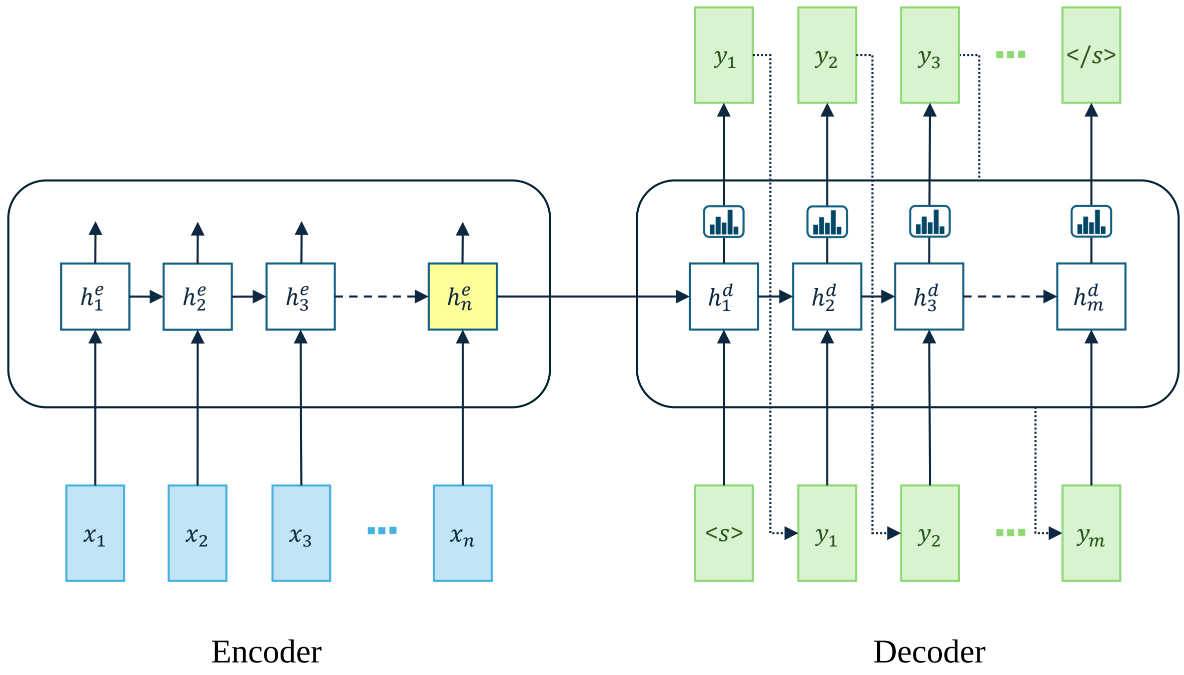


Figure 6: Illustration of the Sequence-to-Sequence (Seq2Seq) model architecture, consisting of an encoder (*left*) and decoder (*right*) model. The last hidden layer of the encoder constitutes the input to the decoder, while the remaining output is ignored during encoding. Adapted from Jurafsky and Martin (2023).

the encoder maps input representations $(x_1, ..., x_n)$ to a sequence of contextualized representations $(h_1^e, ..., h_n^e)$, which are then transformed to an output sequence $(y_1, ..., y_m)$, one element at a time, with n being the length of the input text and m the length of the output text (Jurafsky & Martin, 2023). Both networks are trained end-to-end, which means that they can be trained simultaneously through the provision of parallel data, which contains both the input and the complementary output text, without the necessity of extensive preprocessing steps.

A challenge encountered by Seq2Seq models is the intrinsic information bottleneck, which originates in the restricted amount of information conveyed in the final hidden state h_n^e of the encoder. This limitation restricts the amount of information that can be accessed by the decoder network (Jurafsky & Martin, 2023). This limitation is overcome through the incorporation of direct connections from each of the decoder blocks to the encoder output, an approach referred to as attention mechanism. To determine which elements of the input the decoder should prioritize during the generation process, an attention score is calculated (e.g., as a dot product) between the decoder's hidden state h_{i-1}^d and each encoder's hidden state h_j^e, $j = 1, \ldots, n$ at each generation step $i = 1, \ldots, m$. These attention scores are then converted to a vector of weights $\alpha_{i,j}$ using the softmax function, which are applied to a weighted sum over the encoder states to calculate a context vector $c_i = \sum \alpha_{i,j} h_j^e$, which serves as input to the decoder at each generation step i. The

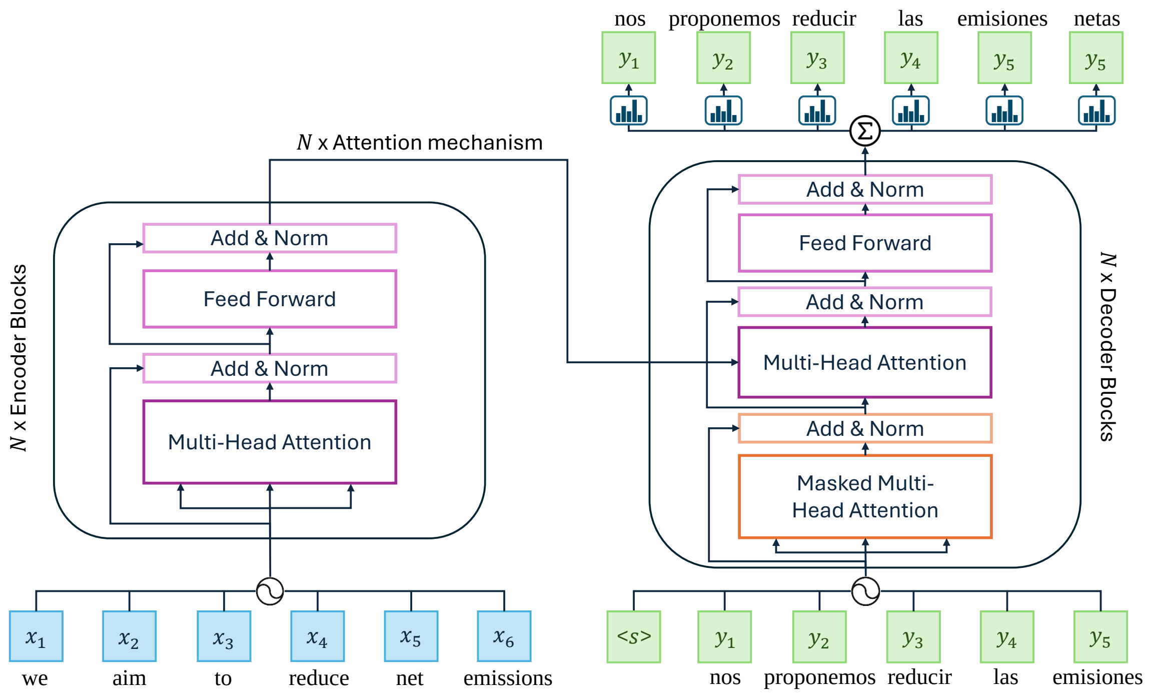


Figure 7: Illustration of the transformer model architecture, consisting of an encoder (*left*) and a decoder (*right*), connected by an attention mechanism. Own illustration adapted from Vaswani et al. (2017).

self-attention mechanism extends the attention mechanism, by additionally computing attention scores on the previously generated output representations at each time step (Jurafsky & Martin, 2023).

The transformer architecture (Vaswani et al., 2017) draws upon the attention mechanism to the point of dispensing with recurrence. This approach enables accelerated training times, as RNNs require sequential training, which is not parallelizable. The transformer architecture consists of transformer blocks, which are composed of a multi-head attention layer and a fully-connected feed-forward network to add non-linearity. The multi-head attention layer performs the attention mechanism h times in h different vector space representations of the encoder and decoder hidden states, thereby enabling the model to jointly attend to different parts of the input via different representation subspaces (Vaswani et al., 2017). The self-attention mechanism is applied both within the encoder side and the decoder side of the model. During training, the decoder is permitted to self-attend exclusively to the states up to the current step of generation, thus preventing the model from attending to text that has not yet been generated. In addition to accelerated training, transformers also facilitate improved explainability, as the attention scores yield information about the internal processes occurring in the model. Figure 7 illustrates the architecture of a transformer model. Figure 8 additionally illustrates a simplified attention distribution, obtained by a self-attention layer within a transformer model.

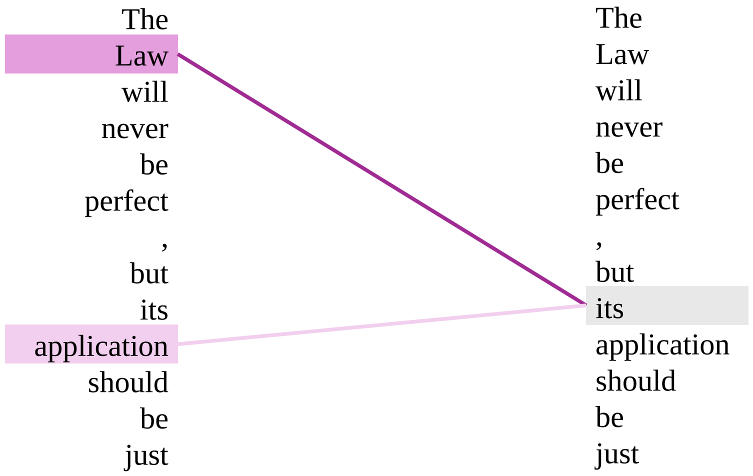


Figure 8: Simplified illustration of the attention distribution obtained by a self-attention layer within a transformer model. The attention distribution indicates, which part of the input the model attends to at each step of the generation process. Own illustration adapted from Vaswani et al. (2017).

Pretrained Language Models

Transformers have been demonstrated to perform well in machine translation, and have also been shown to possess strong generalization capabilities, for instance, as syntactic parsers (Vaswani et al., 2017). These task-agnostic properties of transformers have led to the development of a novel training methodology, known as the “pre-training and fine-tuning paradigm” (W. X. Zhao et al., 2023). This is an application of Transfer Learning (TL) methodology, which involves transferring trained models to a new application area that is related to the tasks they were initially trained for. This can enhance the training process for the target task (Petangoda et al., 2021). The pre-training and fine-tuning paradigm consists of building a language model that obtains general syntactic and semantic understanding capabilities on a task-unspecific text corpus (pre-training) and then train these models on a task-specific corpus to adapt them to specific tasks (fine-tuning).

The pre-training of a language model is typically self-supervised, meaning that the language model does not require labeled data sets for pre-training (Dai & Le, 2015). Pre-training methods differentiate between models that involve the encoder, the decoder, or the entire Seq2Seq model. Decoders perform the task of language modeling, predicting the subsequent word, as illustrated in Figure 9. Consequently, they are trained as a classifier on the last word’s hidden state and perform the tasks pertaining to NLG. An example

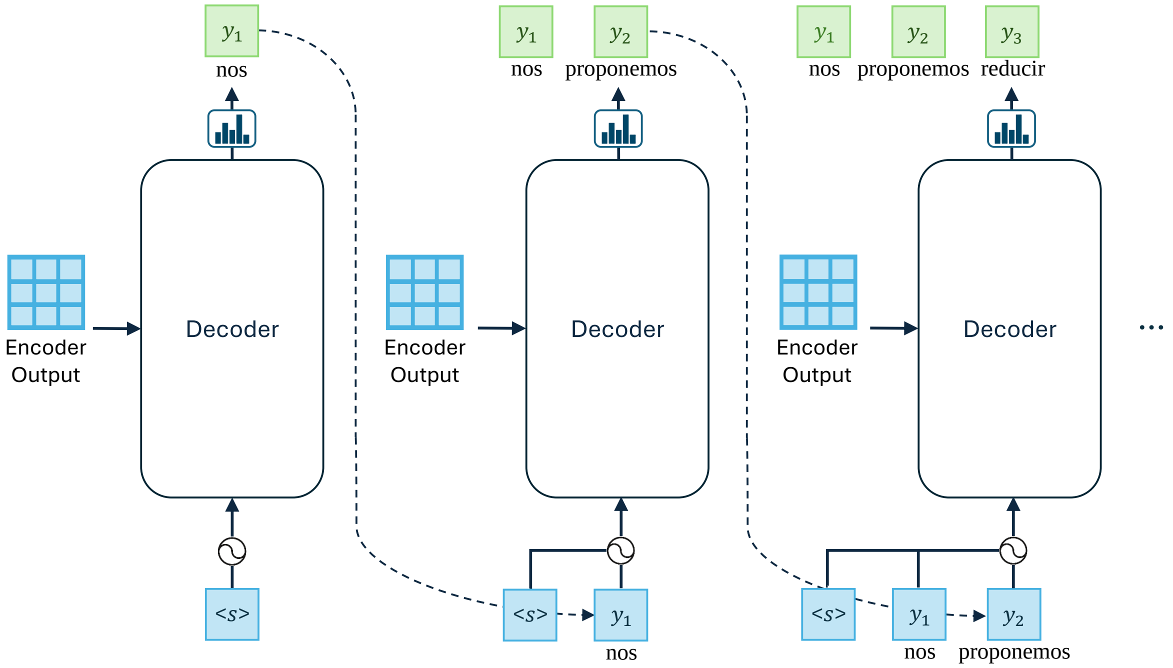


Figure 9: Illustration of a decoder model, performing language modeling. Own illustration adapted from Z. Wang (2023).

of a transformer-based decoder model is the Generative Pre-trained Transformer (GPT) model (Radford & Narasimhan, 2018), which was trained on the BooksCorpus (Zhu et al., 2015) data set.

Encoder models can access entire sentences during pre-training, which makes them suitable for NLU tasks such as NER or text classification. Their learning task involves the process of "masking", which refers to the hiding of individual words (Devlin et al., 2019), which represents an example of self-supervised learning (see Section 2.1.2). Additionally, the encoders receive bidirectional context, which enables them to predict the missing word by utilizing information from both sides of the sentence. The Bidirectional Encoder Representations from Transformer (BERT) model was trained using this approach, leveraging the BooksCorpus and English Wikipedia (Devlin et al., 2019). Various extensions of the BERT model exist, such as SpanBERT which applies masking to a span of words (Joshi et al., 2020). Other examples include RoBERTa (Liu et al., 2019) and DistilBERT (Sanh et al., 2020). The basic masked learning task is illustrated in Figure 10.

Pre-training of encoder-decoder models entails the entire transformer architecture. The model performs the task of language modeling while assessing the training data in a bidirectional manner. Consequently, this kind of model is applied to NLG tasks that are conditional on the input and therefore require NLU capabilities, such as summarization or machine translation. The T5 model (Raffel et al., 2019) and the Bidirectional and Auto-Regressive Transformer (BART) model (M. Lewis et al., 2019) are pre-trained with

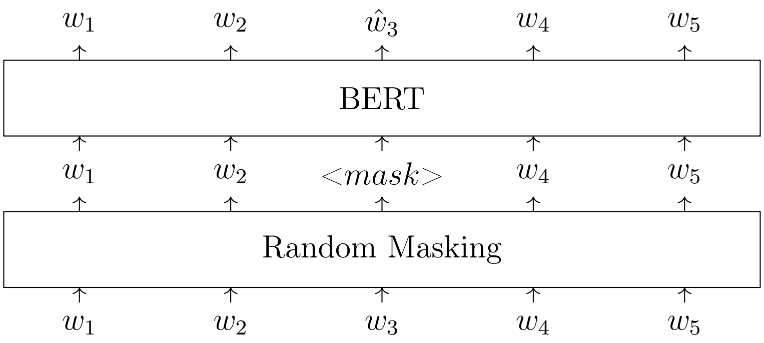


Figure 10: Illustration of masked language modelling task, used to train encoder models. The original sentence $(w_1, \ldots, w_5)$ is randomly masked by replacing a random word w_i with a mask token. The trained encoder model (BERT) predicts the missing word $(\hat{w}_i)$, processing the context from both sides of the masked word. Own illustration adapted from Torregrossa et al. (2021).

an amplified masking technique, which conceals a span of words without specifying the length of the missing part to be predicted.

The fine-tuning task involves selecting a pre-trained model that is well-suited to the task at hand and transferring the model's inherent general syntactic and semantic understanding to a specified fine-tuning task, which is often referred to as the downstream task. This is accomplished by utilizing the architecture and trained weights as an initial point for subsequent training on a downstream task-specific training set. In the case of pre-trained encoder models, a neural network classifier is incorporated on top of the model, which accepts the final hidden state of the encoder as input, to perform the desired downstream task, such as named entity tagging or text classification (Jurafsky & Martin, 2023).

Large Language Models

The latest stage of language models consists of Large Language Models (LLMs), which are largely scaled PLMs (W. X. Zhao et al., 2023). The capacity to perform language modeling, predicting subsequent tokens based on preceding ones improves with model size, enabling LLMs to be employed in a growing number of ways. With the GPT-3 model, pre-trained on 175 billion parameters, Brown et al. (2020) show that these PLMs models are capable of competing with fine-tuned models on downstream tasks, including news article generation, MT and QA. These gained abilities by LLMs through increased model size are referred to as *emergent abilities* (Wei, Tay, et al., 2022).

The integration of examples to facilitate the LLM's comprehension of the desired output, is a technique called *in-context learning* (Olsson et al., 2022) or *few-shot learning* (Brown et al., 2020). It leverages the emergent ability of LLMs to learn during inference, without the necessity of respective gradient steps during training. The ability to perform tasks based solely on instructions provided in the context is referred to as *instruction following* or *zero-shot learning* (Wei, Bosma, et al., 2022). This phenomenon is observed even in cases where the task in question involves multiple steps of reasoning. The emergent ability to infer intermediate reasoning steps from the input context in order to solve complex arithmetic, commonsense, and symbolic reasoning tasks is named *multi-step reasoning* (Wei, Tay, et al., 2022).

The surging of these task-agnostic models demonstrates capabilities that exceed the tasks previously described for NLU and NLG. Instead, the focus in NLP research has shifted from language modeling to encompass a broader range of *task-solving* applications (W. X. Zhao et al., 2023). The successors of the GPT-3 model, such as OpenAI's GPT-4 (Achiam et al., 2024), Google's Gemini (Anil et al., 2024) or Anthropic's Claude models (Anthropic, 2024), were trained on a parameter count estimated to be in the trillions and exhibit additional multimodal capabilities, applying the transformer technology on non-textual data, such as images (Jin et al., 2024). This paves the way for an even broader field of application tasks.

2.2.3 Prompt Engineering

The novel capabilities inherent in LLMs represent a significant advancement in the field of research, which has been designated as *prompt engineering* (Saravia, 2022). A *prompt* is the context provided as input to the LLM. In addition to context information, prompts can also contain task instructions, input data, and output format indications. Consequently, prompt engineering is the process of creating prompts that elicit the desired response from an LLM. This can be achieved either through the process of parameter tuning or by the application of prompting techniques. The following sections will provide further descriptions of both approaches.

Parameter Tuning

The configuration of available LLMs is possible through a series of parameters, which determine the manner in which the model responds to prompts. The *temperature* T influences the softmax distribution (cf. Eq. 2.1), by either flattening the distribution over possible output tokens z_i through a high value of T or by concentrating the mass on the mode of the distribution through a low value of $T- > 0$ (Salamone, 2021):

$$softmax(z_i/T) = \frac{e^{z_i/T}}{\sum_{j=1}^{K} e^{z_j/T}}, \quad for\ i = 1, 2, \ldots, K. \tag{2.7}$$

The latter leads to a low-variance distribution, resulting in a greedy and quasi-deterministic behavior of the model, while highly tempered models favor variability in the generated output, which can be desired in NLG tasks requiring creativity.

A different parameter that allows for the control of variability in output generation is the p-parameter, also referred to as $top - p$. Instead of favoring probable outputs through the redistribution of probability mass, this parameter excludes less probable outputs. This is achieved by limiting the output distribution to the output tokens comprising the top p cumulative mass, thereby defining a top p vocabulary $V(p) \subset V$ of output tokens. This is the smallest set of tokens in the entire vocabulary V such that the cumulative probability of the tokens in the vocabulary

$$\sum_{x \in V^{(p)}} P(x \mid x_{1:i-1}) \geq p \quad for\ i = 1, 2, \ldots, K \tag{2.8}$$

is at least p. This method is also referred to as *nucleus sampling*, as the top p vocabulary is named the *nucleus* (Holtzman et al., 2020).

Further parameters limit the length of the output by defining a upper bound or a *stop sequence* of tokens. Parameters such as the *frequency penalty* or the *presence penalty* apply to penalize repetitions of output tokens, favoring diversity in the vocabulary (Saravia, 2022).

Prompting Techniques

The development of elaborated prompts to instruct LLM to perform tasks with the desired output has resulted in a number of prompting techniques. The following overview is limited to those techniques that are relevant to this work. For a comprehensive overview of the extensive array of prompting techniques that have been developed and elaborated upon in recent years, please refer to the detailed list provided by (Saravia, 2022).

The baseline prompting technique, where the instruction is provided to the model without further examples or demonstrations, is called *zero-shot prompting* (Saravia, 2022). The abilities of LLMs to perform NLP tasks such as text classification through zero-shot prompting are improved by *instruction tuning*, the fine-tuning of LLMs with natural language instructions (Wei, Bosma, et al., 2022). *Few-shot prompting* employs the observed behaviors of LLMs to learn in-context during inference time. In this technique, one or more examples of instructions with corresponding desired output are provisioned in the prompt. This approach assists the models to be guided towards the desired output, resulting in enhanced outcomes compared to zero-shot prompting (Brown et al., 2020). An example of few-shot prompting provided by Brown et al. (2020) is provided in Table 1.

Complex tasks, such as those requiring arithmetic or commonsense reasoning, can be approached by decomposing the original task into several sub-tasks. Two prompting techniques that employ task decomposition to alleviate complexity for the LLM are *Chain-*

Input:	`A "whatpu" is a small, furry animal native to Tanzania. An example of a sentence that uses the word whatpu is:`
	`"We were traveling in Africa and we saw these very cute whatpus"`
	`A "yalubalu" is a type of vegetable that looks like a big pumpkin. An example of a sentence that uses the word yalubalu is:`

Output:	`"I was on a trip to Africa and I tried this yalubalu vegetable that was grown in a garden there. It was delicious."`

Table 1: Example of few-shot prompting, providing a single example to a GPT-3 model. Adapted from Brown et al. (2020)

of-Thought (CoT) (Wei et al., 2023) and *prompt chaining* (Wu et al., 2022). In CoT, the overall task is separated into individual reasoning steps, which yield intermediate results for the reasoning process. These intermediate results are provided within the examples in a few-shot prompt. An alternative to few-shot CoT is zero-shot CoT, which refers to the additional instruction to *"think step by step"*. This has been observed to elicit the multi-step reasoning behavior inherent to LLMs, reaching similar results as by decomposing the task within the prompt (Kojima et al., 2023). An example is provided in Table 2.

Prompt chaining separates the task into separate prompts that are sequentially fed to the LLM to guide the generation process. This leads to an interactive process, where the input to the subsequent sub-task can be dynamically modified based on the latest output. The information gained in each step is consecutively aggregated, resulting in complex outputs to the overall task (Trautmann, 2023). Both task decomposing prompting techniques additionally lead to enhanced transparency through the disclosure of intermediate generation steps (Wu et al., 2022).

A prompting technique that combines NLG with IR methods in order to provision the LLM with external knowledge is presented in the *Retrieval-Augmented Generation (RAG)* framework (Riedel et al., 2020). RAG describes the process of utilizing the prompt input as a query for the retrieval of information in a separate document or knowledge base. The retrieved information is then combined with the initial prompt, resulting in a composed input to the model. The process is illustrated in Figure 11. The achieved addition of context information results in a series of benefits for the output generation. It extends the model's knowledge beyond the scope of the training data, thereby enabling the generation of factually accurate content in the output (P. Lewis et al., 2021). The RAG technique therefore also addresses the problem of reliability of generated output, reducing

Input:	`Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?`
	`A: Let’s think step by step.`

Output:	`There are 16 balls in total. Half of the balls are golf balls. That means that there are 8 golf balls. Half of the golf balls are blue. That means that there are 4 blue golf balls.`

Table 2: Example of zero-shot CoT prompting, instructing a GPT-3 model to “think step by step”. Adapted from Kojima et al. (2023).

the amount of hallucination and leading to improved consistency in the output text. This consequently augments the precision and reliability of the generation, particularly for tasks that are knowledge-intensive. RAG further enables the incorporation of continuous updates to knowledge and the integration of domain-specific information. These notable characteristics of RAG have resulted in this technology receiving significant attention over the past years and being employed in a variety of applications, either in its original form or in a modified version. Given that this technology is still in its development phase with the constant appearance of new contributions, recent survey papers such as Gao et al. (2024) have significantly contributed to the field by establishing an initial taxonomy and shaping the definitions of several terms, which have been extended by further studies (P. Zhao et al., 2024) (Hu & Lu, 2024).

2.2.4 Information Retrieval

IR is a research field closely related, and sometimes even regarded as a subfield (Manning & Schütze, 1999), to NLP. It has gained special relevance with the widespread usage of internet searching, as it involves the identification and retrieval of relevant information from a corpus of unstructured data based on user queries (Russell et al., 2022). The development of sophisticated IR systems has been crucial in managing the exponential growth of online information, enabling users to efficiently locate documents, articles, and other resources. With the emergence of RAG approaches that make use of IR techniques, the linkage between the two fields of research has increased even further.

Vector Space Model

In this context, the *Vector Space Model* is of special relevance, which is a widely applied model for IR (Manning & Schütze, 1999). It uses spacial proximity as an indicator for semantic proximity of a data point to the query, and thus also makes use of dense

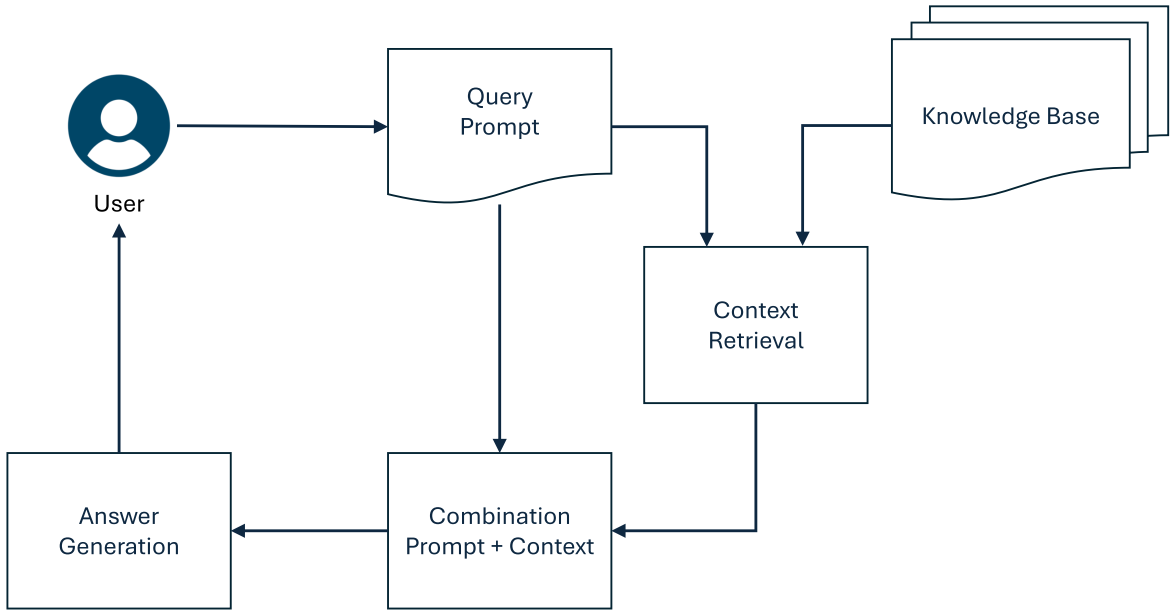


Figure 11: Illustrated RAG framework, adapted from Gao et al. (2024).

representations yielded by text embeddings (cf. Figure 2). The relevance of entries to a query is therefore determined by the values of the calculated similarity measures between the dense representations in vector space, and the results are ranked accordingly.

Vector Similarity

Several methods exist to determine the spacial proximity between vectors. In addition to the Euclidean distance measure, a common technique is to define proximity of vectors by calculating the *cosine similarity*,

$$sim(X,Y) = \frac{\langle X, Y \rangle}{\|X\| \|Y\|} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \sqrt{\sum_{i=1}^{n} y_i^2}} \tag{2.9}$$

which corresponds to the cosine of the angle between the vectors X and Y in n-dimensional embedding space (Manning & Schütze, 1999). In comparison with other metrics, such as the simple dot-product between vectors, this metric yields the benefit that the similarity values are inherently normalized to the range of $[-1, 1]$, enhancing the interpretability of the resulting similarity scores.

The Term Frequency–Inverse Document Frequency (TF-IDF) measure (Salton & Buckley, 1988) is an extension of the cosine similarity. In addition to term frequency (TF), the model also considers inverse document frequency (IDF). This assigns less weight to terms that occur frequently within the language, as they are considered less meaningful.

2.3 Climate Policy

Climate policy refers to the measures and strategies adopted by governments and international bodies to address the challenges posed by climate change. These policies encompass a wide range of actions, from emissions reductions and energy transitions to climate funding and adaptation strategies (Clark et al., 2023). The following section seeks to provide a brief introduction into the topic of climate policy, with emphasis on relevant institutions and documents for this thesis.

2.3.1 Policy

The term *policy* is multifaceted and encompasses various concepts. Public policy, in general, refers to means by in which governments seek to influence behavior and achieve specific objectives through legal instruments and implementation formats such as acts, laws, and regulations (Planas et al., 2021). In the context of climate policy, this involves the creation and implementation of frameworks that support sustainable development and environmental protection, leading to a broad scope of intervention areas, which may also encompass distribution policy, foreign trade policy and the policies of international organizations (Welfens, 2022).

2.3.2 Institutions

Several institutions are involved in shaping and implementing climate policies at national and international level. The UNFCCC serves as the primary international platform for climate negotiations, bringing together governments of UN member states and additional parties, such as the EU, to discuss and agree on measures to combat climate change (Leggett, 2020). All parties to the UNFCCC meet annually in Conference of the Parties (COP) conventions to "evaluate the national communications and emission inventories submitted by Parties" (UNFCCC, 2024a). These COP conventions have been the cradle of milestone treaties such as the Kyoto Protocol and the Paris Agreement (Leggett, 2020). The IPCC is the scientific body of the UN and provides assessment reports (ARs) (Calvin et al., 2023) that conduct a systematic review of the current state of climate research, with the objective of informing the negotiations and guiding policy development (Leggett, 2020).

In addition to the UN, a number of other institutions are involved in the creation, assessment, and influence of climate policy documents. These include national and regional governments, Non-governmental Institutions (NGOs), and private sector entities. It is the responsibility of national governments to develop and enforce domestic climate policies that are aligned with international commitments, such as the Paris Agreement. NGOs contribute to the process through advocacy and research, while the private sector plays a pivotal role in driving innovation and investment in sustainable technologies and practices, thereby facilitating the transition of economies towards low-carbon models (Calvin et al., 2023).

2.3.3 The Paris Agreement

The Paris Agreement entered into force in November 2016 and aims to "holding the increase in the global average temperature to well below 2°C above pre-industrial levels and pursuing efforts to limit the temperature increase to 1.5°C above pre-industrial levels, recognizing that this would significantly reduce the risks and impacts of climate change" (UNFCCC, 2016). In order to achieve this objective, the treaty stipulates two core policy documents to be contributed by each Party to the Agreement, which are outlined in the following section.

Nationally Determined Contributions (NDCs)

NDCs, also referred to as Climate Action Plans (CAPs) are a key component of the Paris Agreement (UNFCCC, 2024b). They represent the efforts by each Party to the Agreement to reduce national emissions and adapt to the impacts of climate change. Within the documents, the parties commit to prepare, communicate, and maintain increasingly ambitious climate actions and plans they intend to achieve. NDCs are not legally binding, but countries are required to adopt domestic measures with the objective to achieve their targets. As a mechanism of auditing, the Paris Agreement established a common framework for reporting and reviewing NDCs to ensure transparency and accountability (Bodansky, 2016).

NDCs vary significantly between countries in terms of their content and level of ambition. Some countries have set economy-wide emission reduction targets, while others have outlined sector-specific actions or qualitative measures. The diversity in NDCs can be attributed to differences in national circumstances, capacities, and priorities, as the treaty recognizes “the need to support developing country Parties for the effective implementation of this Agreement” (UNFCCC, 2016). To enhance the comparability and effectiveness of NDCs, the Paris Agreement includes guidelines for their preparation, including the provision of quantitative information on targets, reference values, implementation periods, sectors covered, and assumptions used (Bodansky, 2016).

A key mechanism to ensure climate ambition is the continuous renewal of NDCs. This is carried out in the context of a global stock take performed every five years to assess collective progress towards achieving national targets and global goals (UNFCCC, 2016). While the targets and measures of NDCs are not legally binding, their communication and successive enhancement on a five-year basis are (Bodansky, 2016). A Party may additionally adjust its existing NDC at any time with enhancements to its level of ambition (UNFCCC, 2016).

At the time of their introduction in 2015, no fixed specifications were established concerning the structure and content of NDCs, which were at that time referred to as *Intended* NDCs (INDCs). Consequently, these document present considerable diversity, particularly in the initial submissions. Instead, templates provided by external organizations, such as Climate and Development Knowledge Network (CDNK) (Wartmann et al., 2023), illustrated in Figure 3, serve as a basis for the creation of more standardized structures. Since 2020, requirements regarding the content have been adopted, initially proposed in Decision 4/CMA.1 at COP 2018 (UNFCCC, 2018), which serve as guidance and are "strongly encouraged" (Casas et al., 2021). They contain the following aspects:

1. Quantifiable information on the reference point (including, as appropriate, a base year)
2. Time frames and/or periods for implementation
3. Scope and coverage
4. Planning processes
5. Assumptions and methodological approaches, including those for estimating and accounting for anthropogenic greenhouse gas emissions and, as appropriate, removals
6. How the Party considers that its NDC is fair and ambitious in the light of its national circumstances
7. How the NDC contributes towards achieving the objective of the Convention

Long-Term Low Emission Development Strategies (LT-LEDS)

The Paris Agreement additionally encourages countries to voluntarily develop and communicate Long-Term Low Emission Development Strategies (LT-LEDSs), also referred to as Long-Term Strategies (LTSs). These policy documents guide the countries' climate action by stating longer-term visions for the countries' development and typically include targets for mid-century emission and decarbonization pathways for key sectors such as energy, transport, and industry. Consequently, they are intended to complement the NDCs, which delineate immediate action and targets on a shorter-term horizon (e.g., 10-year timeline), while linking short-term planning to long-term objectives (UNDP, 2023).

National Context	This section provides the overall national context for the INDC, including how the actions set out in the INDC fit with national sustainable development priorities and existing plans and strategies. It can also contain a headline summary of the INDC as a whole.
Mitigation	**Contribution:** This section contains a summary of the mitigation contribution, including type of contribution, level of ambition and any conditionality that may be relevant for the contribution. It should be noted that countries may wish to specify a long-term outcome (e.g. up to 2050) as well as short-term outcome for the period to 2025 or 2030. **Information to facilitate clarity, transparency and understanding:** This section includes detailed information to improve understanding of the contribution and allow comparability with other contributions. **Fair and ambitious:** This section sets out how the contribution is considered to be fair and ambitious in light of the country's national circumstances and the objective of the UNFCCC set out in Article 2.
Adaptation	This section provides an opportunity for countries to highlight current and future adaptation action including adaptation-mitigation synergies, as well as the support that may be required for implementation of adaptation plans, developing capacity or scaling up interventions.
Planning Process	This section provides an overview of existing or planned domestic processes for monitoring and supporting the implementation of the INDC.
Means of Implementation	This section describes the financial, capacity-building, technology transfer or other types of international support related to the INDC; this information may help international partners to better understand and target their support.

Table 3: NDC template proposed by CDNK (Wartmann et al., 2023).

3 Related Work

The advanced techniques in the fields of ML and especially NLP presented in Chapter 2 have incentivized a range of studies applying them in the context of policy analysis. The recent attempts to evaluate the potential of NLP in facilitating policy analysis have encompassed a wide range of policy domain, such as forestry (Firebanks-Quevedo et al., 2022), food waste (Aitken et al., 2022), environmental protection (Cação et al., 2021) and energy (Carroll et al., 2024). These developments have caused Safaei and Longo (2023) to question whether human involvement is necessary in policy analysis at all. To this end, a GPT-2 model was fine-tuned on policy documents to create briefing notes based on them. The objective of their experimental study was to compare three different versions of briefing notes based on public policy documents. In addition to the purely automated version, a human-generated version was presented, as well as an automatically generated and human-reviewed version. The three versions were evaluated by human expert judges, and the experiment was conducted for two policy domains that were thematically distinct. Even though the automated briefings were not able to entirely convince the judges in terms of plausibility, persuasiveness, and usefulness, the study represents a first approach towards a deeper understanding of automation potential in the policy analysis domain. Further, the findings of the study may already be superseded by the relentless introduction of enhanced LLMs (W. X. Zhao et al., 2023), which surpass the abilities of the GPT-2 model applied by the authors.

The following chapter seeks to provide an overview of further research addressing the application of NLP in the climate domain. In particular, it presents a review of previous studies that have employed NLP techniques for the analysis of climate policy documents. The discussed literature is additionally summarized and presented in Tables 4 and 5 at the end of the chapter.

3.1 NLP and Climate Change

As the scientific and political focus on climate change continues to expand, so do efforts to utilize novel methodologies from the domains of ML and NLP to limit and address the consequences of climate change. Rolnick et al. (2023) provide an extensive overview of the potential applications of ML and NLP in problem-solving across a range of climate change-related domains, such as energy systems and transportation. The article identifies areas where these techniques could be highly impactful both in the short and in the long-term.

With regard to NLP methods, recent studies have particularly intensified their application in the analysis of climate change discourses and discussion through a variety of channels, including social media and political platforms (Stede & Patz, 2021). Hätönen and Melzer (2021) propose a hybrid method, combining word embeddings and topic modeling, to monitor political discussions, to assess when and how politicians comment on the topics of climate change and climate policies, analyzing social media posts, as well as transcripts from political speeches and blog entries. Grundmann (2022) reviews the application of similar methods to analyze newspaper articles on climate change in a semi-automated manner. Further research has dealt with tweets and the question whether climate change is linked to an increase in hate speech (Stechemesser et al., 2022).

G. Wang et al. (2021) develop a climate claim fact-checking pipeline by fine-tuning a RoBERTa model to detect and extract climate claims from text samples and query reliable websites for evidence. A second fine-tuned RoBERTa model is leveraged to decide whether the retrieved evidence confirms the claim or exposes it as a false claim. The proposed pipeline is applicable to larger texts such as newspaper articles, as well as isolated claims in social media, such as tweets.

Similar approaches utilizing PLMs to adapt them to climate-related use cases are numerous. Specially, the BERT model has been applied in a variety of contexts to perform specific downstream tasks.

Kölbel et al. (2022) fine-tune a BERT model on climate-related text to apply it to climate risks disclosures provided by companies to inform about their exposure to risks related to climate change. The model is intended to identify and distinguish both physical risks that emerge from extreme weather events and transition risks stemming from regulatory reforms intended to combat climate change, and outperforms other approaches such as simple BOW models. In similar settings, Varini et al. (2021) fine-tune a BERT model by adding a classification head to detect climate-related phrases in risk disclosure documents, whereas Luccioni et al. (2020) fine-tune a RoBERTa model with the same task on financial and sustainability reports.

Callaghan et al. (2021) fine-tune a DistilBERT model on three separate tasks. The first model serves as a binary classifier to detect relevant documents describing climate impacts. A second fine-tuned DistilBERT model performs the downstream task of multi-labeling a set of impact categories on the retrieved relevant documents. A third and final model defines the driver of the climate impact to be either precipitation or temperature rise. All three fine-tuned PLMs outperform respective SVMs trained for the same tasks.

Given these numerous applications of PLMs in the climate sector, some approaches go one step further and propose to additionally pretrain these models to handle the specific vocabulary and nuances of climate-related texts, making them more effective in identifying and classifying climate-related content. This technique is referred to as domain-adaptive pretraining (Gururangan et al., 2020).

Webersinke et al. (2022) proposed to domain-adapt a base DistilRoBERTa model (Sanh et al., 2020) on a climate-related text corpus containing various sources, including news articles, corporate disclosures and scientific articles. This model named ClimateBERT was evaluated on several downstream tasks, including a text classification task to detect climate-related paragraphs, sentiment analysis and fact-checking. The model outperforms the DistilRoBERTa base model in all three use cases. Jalalzadeh Fard et al. (2022) propose to amplify this approach by further including policy and scientific documents from the health sector to a multi-domain-adaptive pretraining set.

A more recent domain-adapted model is ClimateGPT-7B (Thulke et al., 2024), a decoder model based on Llama-2 (Touvron et al., 2023) that was pretrained on a collection of scientific climate documents and reports, to adapt to the climate domain. The model is specifically designed for climate-related QA-tasks in the context of a homonymous RAG application[1]. The results of automated and human evaluation show that the 7B-domain-adapted model competes on par with the 10-times larger 70B-Llama2 base version and outperforms the same-size benchmark that was used as base model. Thulke et al. (2024) also apply a promising cascading machine translation approach to face multi-linguality limitations of the English-native base model. A similar climate-dedicated QA-machine called chatClimate[2] has been developed by Vaghefi et al. (2023). The conceived application uses GPT-4 as base model and applies a RAG approach to leverage information contained in IPCC AR6 reports (see Chapter 2.3.2) to provide grounded knowledge for the answer generation.

3.2 NLP in Climate Policy

The application of NLP and ML can impact the way to approach climate policy documents on various levels, as detailed by Rolnick et al. (2023). From the policymaking perceptive, these techniques may facilitate the provision of data for climate policy drafting, lead to the creation of novel tools for policy assessment and enhance the efficacy of policy evaluation. From the research or citizen perspective, NLP methods provide the opportunity to facilitate the digestion of lengthy and complex laws and policy documents and therefore grant the ability to engage with climate policy decisions, by presenting the contained information in a manageable and understandable manner. The following section is intended to provide an overview of recent studies that have applied NLP methods to extract information relevant to either end.

Abudu et al. (2024) focus on climate bond policies, applying topic modeling to detect contained policy objectives. Their results highlight a shift in fund allocation from energy

[1] ClimateGPT: https://www.climategpt.ai/, retrieved July 23, 2024

[2] chatClimate: https://www.chatclimate.ai/, retrieved July 23, 2024

projects to construction work, while a sentiment analysis discloses skeptical views on climate change contained in some of the processed documents, raising concerns about the effectiveness of stated actions. A similar approach is applied by Żółkowski et al. (2022), who apply text modeling on National Energy and Climate Plans (NECPs) of EU member states, followed by clustering of countries' topic distribution, to detect alignments in policy framing.

Swarnakar and Modi (2021) propose a generally applicable framework for an integrated knowledge platform that would extract relevant information coming from structured and unstructured sources, including but not restricted to climate policy documents, and store it in a dedicated database. Discussed use cases contain topic modeling, sentiment analysis and discourse analysis. A similar framework is proposed by Planas et al. (2021), who describe an integrated end-to-end pipeline which entails a variety of NLP tasks, including summarization, NER and text classification utilizing a Sentence-BERT model (Reimers & Gurevych, 2019). The overarching objective of the authors is the establishment of a Knowledge Graph (KG), which is a structured semantic data model that outlines the entities and relationships retrieved in the individual modules of the framework.

A substantial number of studies addressing climate policy documents have employed climate domain-adapted models described in chapter 3.1. Clark et al. (2023) propose a pipeline that predicts voting outcomes for climate legislation by applying a fine-tuned ClimateBERT or CliMedBERT model. The objective of the authors is to gain an increased understanding of voting outcomes by employing the models, in order to improve transparency and to assist policy experts in the process of legislative advocacy.

Sietsma et al. (2022) fine-tune two ClimateBERT models with the aim of creating a global evidence map on climate change adaptation policies. For this purpose, the first model is fine-tuned as a binary classifier to label relevant policies from a database. A second model performs a fine-grained multi-label classification to detect adaptation topics treated in the documents.

ClimateBERT-NetZero is a fine-tuned ClimateBERT model for the detection of net-zero emission and emission reduction targets in sustainability reports of private and public institutions (Schimanski et al., 2023). The classification head is trained on data provisioned by the Net Zero Tracker project[3]. Analogous to this, Juhasz et al. (2024) extract the same type of information from national policies and UNFCCC submissions by leveraging data contained in the Climate Policy Radar[4] to fine-tune ClimateBERT. The authors additionally apply topic modeling with BERTopic (Grootendorst, 2022) to assign sectors to extracted paragraphs.

[3] Net Zero Tracker: https://www.zerotracker.net/, retrieved July 23, 2024

[4] Climate Policy Radar: https://app.climatepolicyradar.org/, retrieved July 23, 2024

Corringham et al. (2021) assign topics to sentences from NDCs by fine-tuning a BERT model. The topic labels of the training set were previously inferred from the documents' headers, as alternative to human expert annotation. The performance of the model is evaluated in comparison to a baseline BOW model, which was previously employed by Biesbroek et al. (2020) to a similar task of classifying "adaptation" and "mitigation" sections in national policy documents. Spokoyny et al. (2023) test ClimateBERT among other models to build a collection of QA-datasets from unstructured policy documents, such as NDCs, by transforming them into semi-structured questionnaires. The contained question answer pairs serve as training and evaluation data for climate-domain QA-tasks (Thulke et al., 2024).

Sustainable Development Goals (SDGs) are the subject of work by Pukelis et al. (2022). Their OSDG 2.0 tool[5] is capable to detect SDG-relevant content and assign suggested SDG-labels in text input. The classification task is performed by traditional ML classifiers, such as multinomial regression models, trained on community labelled data[6]. The initial labelling is further tested for consistency with an ontology map conceived for the same task (Pukelis et al., 2020).

A number of text classification models for climate policy analysis have been released by German Development Cooperation (GIZ). The Policy Action Tracker[7] was built in collaboration with United Nations Sustainable Development Solutions Network (SDSN) and consists of a contextual search functionality, which allows querying policy documents with keywords or questions and retrieves corresponding text from the document by calculating similarity on the level of dense representations. It additionally employs the OSDG 2.0 tool to identify pertinent information related to SDGs in the document and provides a comparison with the country's claims in the respective NDC. This comparison serves to establish a foundation for evaluating the coherence between policy claims and their implementations, and is performed based on data provided by the German Institute of Development and Sustainability (IDOS)'s NDC Explorer[8].

A Sentence-BERT model is fine-tuned for the classification task of detecting references to a list of pre-defined groups in vulnerable situations with regard to climate change for a variety of reasons, such as age, socio-economic status or geography (Voigt et al., 2024). The model is wrapped by the Vulnerability Analysis App[9], which enables its application

[5] OSDG 2.0: https://www.osdg.ai/, retrieved June 14, 2024

[6] OSDG Community Dataset (OSDG-CD): https://www.github.com/osdg-ai/osdg-data, retrieved June 14, 2024

[7] Policy Action Tracker: https://www.huggingface.co/spaces/GIZ/SDSN-demo, retrieved June 14, 2024

[8] IDOS NDC Explorer: https://www.klimalog.idos-research.de/ndc/#NDCExplorer, retrieved June 14, 2024

[9] Vulnerability Analysis App: https://www.huggingface.co/spaces/GIZ/cpv_3, retrieved June 14, 2024

to climate policy documents for the purpose of determining whether the respective policy document addresses these groups and, if so, to what extent.

In summary, the studies and literature presented in this chapter largely emphasize the adaptation of general-purpose PLMs through domain-specific pretraining or fine-tuning for climate-related downstream tasks. To the author's knowledge, however, there is a lack of research examining the extent to which state-of-the-art LLMs, trained on a broad body of knowledge, are directly applicable to analyze climate policy documents. This could signify a major improvement in terms of practicability, as the efforts related to pretraining and fine-tuning would be eliminated. Moreover, it is anticipated that this approach could have a positive effect on the environmental impact of the process, as training LLMs is associated with high energy consumption and thus, depending on the energy source, leaves a substantial carbon footprint (Strubell et al., 2019).

Towards this intent, the thesis draws inspiration from the work of Trautmann (2023), who employs the prompt chaining technique with general-purpose LLMs for the classification of legal documents. These documents exhibit a number of similarities with climate policy documents, with regard to their considerable length, complex language and domain-specific vocabulary. While being less resource-intensive in terms of time and energy, the application of general-purpose models via prompting has proven to be competitive with fine-tuned models in climate-related classification tasks (Schimanski et al., 2023) as well as other NLP tasks such as MT (Robinson et al., 2023). Other experiments have shown that general purpose models even outperform domain-specific models in the climate sector (Spokoyny et al., 2023).

Category	Focus	Task	Fine-Tuning	Domain-Adaptive Pretraining	LLM Prompting	Source
NLP in Policy Analysis	Food Waste	Clustering	□	□	□	Aitken et al. (2022)
	Foresty	Classification	■	□	□	Firebanks-Quevedo et al. (2022)
	Energy	Classification	□	□	□	Carroll et al. (2024)
	Environmental Protection	Classification	■	□	□	Cação et al. (2021)
	Policy Briefing	Text Generation	□	□	■	Safaei and Longo (2023)
	Legislation	Classification	□	□	■	Trautmann (2023)
NLP in Climate Change	General	Overview	□	□	□	Rolnick et al. (2023)
	Media Discourse	Overview	□	□	□	Stede and Patz (2021)
	Polit. Discourse	Topic Modeling	□	□	□	Hätönen and Melzer (2021)
	News Arcticles	Overview	□	□	□	Grundmann (2022)
	Hate Speech	Classification	□	□	□	Stechemesser et al. (2022)
	Fact-Checking	Classification, IR	■	□	□	G. Wang et al. (2021)
	Risk Disclosure	Classification	■	□	□	Kölbel et al. (2022)
	Risk Disclosure	Classification	■	□	□	Varini et al. (2020)
	Sustainability Reports	Classification	■	□	□	Luccioni et al. (2020)
	Climate Impact	Classification	■	□	□	Callaghan et al. (2021)
	General	Classification	■	■	□	Webersinke et al. (2022)
	General	QA, MT	■	■	□	Thulke et al. (2024)

Table 4: Overview of related work regarding NLP in Policy Analysis and Climate Change. Applies: ■ ; not applies: □ .

Category	**Focus**	**Task**	**Fine-Tuning**	**Domain-Adaptive Pretraining**	**LLM Prompting**	**Source**
	General	Classification, NER	■	■	□	Jalalzadeh Fard et al. (2022)
	General	QA	□	□	■	Vaghefi et al. (2023)
NLP in Climate Policy	Climate Bonds	Topic Modeling, Classification	□	□	□	Abudu et al. (2024)
	EU NECP	Topic Modeling, Clustering	□	□	□	Żółkowski et al. (2022)
	Knowledge Platform	Topic Modeling, Classification	□	□	□	Swarnakar and Modi (2021)
	Knowledge Platform	Classification, Text Generation	■	□	□	Planas et al. (2021)
	Legislation	Classification	■	■	□	Clark et al. (2023)
	Adaptation Policy	Classification	■	■	□	Sietsma et al. (2022)
	Sustainability Reports	Classification	■	■	□	Schimanski et al. (2023)
	Emission Targets	Classification, Topic Modeling	■	■	□	Juhasz et al. (2024)
	NDCs	Classification	■	■	□	Corringham et al. (2021)
	NDCs	QA	■	■	□	Spokoyny et al. (2023)
	SDGs	Classification	□	□	□	Pukelis et al. (2022)
	NDCs	Classification	■	□	□	Voigt et al. (2024)

Table 5: Overview of related work regarding NLP in Climate Policy Analysis.
Applies: ■ ; not applies: □ .

4 Concept

The vast majority of the presented studies and literature in Chapter 3 share the common approach of leveraging existing general PLMs to either perform domain-specific pretraining to the climate domain or to fine-tune the pretrained models to a climate-related downstream task. The present thesis is guided by the following research questions:

- To what extent can the advent of general-purpose LLMs, along with the accompanying performance gains due to their emergent abilities (see Chapter 2.2.2), be leveraged directly for the task of climate policy document analysis, thus circumventing the need for additional costly processes such as pretraining or fine-tuning?
- Is it possible to automize the analytical process through the application of LLMs?

A conceptual framework for an analytical end-to-end pipeline is developed to address these questions. The following chapter offers a comprehensive explanation of the selected methodology and the concept that was conceived for this task. To this end, the chapter begins with a specification of the analysis task and the resulting challenges and requirements. This is followed by a detailed description of the designed pipeline concept and its components.

4.1 Task Description

The pipeline proposed in this chapter is designed for a specific task within the field of climate document analysis. It consists of extracting targets and measures from the documents, which are the contents that contain specific claims by the entities issuing the document.

In accordance with the definition employed by Juhasz et al. (2024), in the context of this thesis, a climate policy target is characterized by three criteria:

1. A target specifies a desired outcome or objective,
2. it is quantifiable or qualitatively measurable, and
3. it is time-bound by stating a deadline or target year for the achievement of the target.

This definition also aligns with the SMART (Specific, Measurable, Ambitious, Relevant, Time-bound) concept for climate policy targets (Wartmann et al., 2023), which is specifically designed for the evaluation of NDCs targets and sets out ambition and relevance as additional criteria. These aspects are notoriously difficult to evaluate, in particular for LLM classifiers that are not trained for this task, which is why their inclusion would exceed the scope of this thesis.

The distinction between targets and measures lies in the notion that measures are defined as planned actions which are not required to be quantifiable or time-bound. Therefore, this category is less specific and includes a wider range of statements, such as declarations of intent, high-level plans, or even desired outcomes (International Transport Forum, 2018). An alternative approach is defining measures as actions and plans that delineate pathways intended to facilitate the realization of the established targets.

4.2 Challenges and Requirements

The manual and automated analysis of climate policy documents, such as NDCs, presents a number of challenges due to the intrinsic complexity and diversity of such documents. The conceived pipeline has been developed with the specific intention of addressing these challenges.

1. Policy documents are typically extensive and characterized by domain-specific and technical language (Juhasz et al., 2024), including comprehensive descriptions of emissions reduction scenarios, adaptation strategies, financial mechanisms, and implementation timelines. The information is often presented in tables, which is beneficial for manual analysis but adds another layer of complexity to automation (Fang et al., 2024). Extracting and interpreting this tabular data is challenging because it often requires understanding the context provided in the surrounding text. Moreover, tables may vary in format and structure, necessitating advanced techniques for data extraction and normalization to ensure comparability and coherence in the analysis.

2. Another significant challenge in analyzing climate policy documents lies in the subtle packaging of critical information. For example, the conditionality of targets is a common feature in NDCs. Countries may present targets that are contingent upon receiving international support, such as funding or technology transfer. These conditional targets are often embedded within the text in a manner that requires careful scrutiny to identify and interpret. Differentiating between unconditional and conditional targets is crucial for assessing the genuine commitment and feasibility of a country's climate action plans.

3. National policy documents are published in the official languages of the respective countries. This results in a vast array of languages, requiring accurate translation and understanding of climate-specific terminology across each of them.

4. In addition to these factors, climate policy documents are dynamic in nature, subject to updates and revisions as countries enhance their commitments and strategies. The evolving nature of these documents necessitates continuous monitoring and reanalysis to ensure that the latest information is considered in policy assessments. This dynamic quality of these documents further complicates the task of maintaining up-to-date and accurate analyses.

These challenges result in a list of requirements to the elaborated concept. They relate both to the pipeline architecture and to the language models used for text analysis. With regard to the latter, the following requirements are derived:

1. The LLMs contained in the pipeline must be familiar with the domain-specific language employed in the documents and capable of understanding technical terms related to the field of climate policy and the sectors involved.

2. The integrated LLMs are required to be capable of processing information contained in different representations. In particular, the information contained in tabular form must be adequately processed, given the prominent role of tables in climate policy documents.

3. Another required asset of the employed LLMs is multi-linguality, given the diverse languages in which climate documents are issued. In terms of language capabilities, the minimum requirement is the ability to properly process the six official UN languages accepted for UNFCCC submissions, which include Arabic, Chinese, English, French, Russian, and Spanish (United Nations, 2024). Alternatively, an accurate machine translation can be included in the process in order to address language limitations of the applied models, as proposed by Thulke et al. (2024).

The additional requirements to the architecture entail the following aspects:

4. The pipeline must be able to handle the common file types used to submit policy documents. This most commonly includes, but is not limited to, PDF documents. Other permissible formats include Microsoft Word files.

5. In some cases, excerpts, or even entire documents are submitted as scans. It is therefore necessary for the pipeline to have Object Character Recognition (OCR) capabilities, which refers to a technique from the field of computer vision that extracts text from image data. Beyond this, images assume a secondary role in climate policy documents, serving primarily as visual aids in the form of graphics. They are thus not specifically addressed in the scope of this thesis.

6. The pipeline must be capable of analyzing large documents, reaching up to several hundred, and scale in an acceptable manner in terms of cost and runtime.

7. Each target and measure, as defined in Section 4.1, is generally expressed within one or a few sentences (Planas et al., 2021). The basic unit of analysis should therefore be single sentences or short sequences of related sentences. In the following, each unit of analysis is referred to as *quote*. Quotes represent the proof for a target or measure expressed in the document, making them the main output of the pipeline.

8. Documents typically contain multiple targets and measures (Planas et al., 2021). Consequently, the pipeline must be capable of extracting multiple quotes.

9. It is necessary to verify the extracted information, as LLMs are subject to the tendency of making up information. This is a phenomenon referred to as *hallucination*. Consequently, the extracted quotes must be phrased verbatim from the original text.

Table 6 provides an additional overview of the defined requirements.

Aspect	**Challenge**	**Requirement**	**Pipeline Module**
Architecture	Submissions in diverse document formats	Ability to process different document formats	Preprocessing
	Lengthy policy papers reach several hundreds of pages	Acceptable scaling properties	Preprocessing, Retrieval, Quotation, Classification
Architecture/ LLM	Relevant information is contained in individual sentences	Ability to extract quotes from the document corpus	Quotation
	LLMs tend to hallucinate	Verbatim reproduction of the extracted quotes	Quotation
	A single document may contain multiple relevant quotes	Ability to extract various quotes from a single document	Quotation
	National policy papers and UN submissions issued in diverse languages	Multilingual LLM, at least covering the six official UN languages, or alternatively, machine translation step	Preprocessing, Retrieval, Quotation, Classification
	Documents in occasions submitted as scans	OCR capabilities included in the pipeline	Preprocessing
LLM	Domain-specific and technical terminology	LLM familiar with climate domain-specific and technical terminology	Preprocessing, Quotation, Classification
	Relevant information contained in tabular format	LLM is capable of processing tables	Preprocessing, Quotation

Table 6: Requirement list to the architecture and language models applied for automated climate policy analysis.

4.3 General Outline

Based on the specified task and resulting requirements, the selected methodology is presented in the following section.

4.3.1 Pipeline Definition

As in Żółkowski et al. (2022), this thesis applies the definition of pipeline as a set of procedures and methods that are sequentially applied to an input to generate results. These results encompass the final output, as well as intermediate results, such as the extracted text from documents. The additional designation as *end-to-end* pipeline is used to describe a process that is fully integrated, from the initial input of a raw document by the user to the final output of the finished result. This implies that no additional processing steps are required.

The pipeline concept was selected as overarching structure, as the complexity of the task necessitates a sequence of procedures and methods to be applied on the input document. Primarily, the length of the documents may exceed the permitted context length of many LLMs. Additionally, there is a possibility that the LLM may be unable to direct attention on the entirety of relevant information contained, if the document is analyzed in a single step. The integration of the defined sequence of processes within a pipeline should also facilitate automation of the analysis, removing the necessity for additional human intervention.

4.3.2 Automation

The proposed automated end-to-end analytical pipeline is primarily based on the established manual process applied by human analysts for the evaluation of climate policy documents. Following the descriptions by climate policy experts, the sequence of steps involved in the manual process is illustrated in Figure 12.

When confronted with a novel policy document, the analyst initiates the process by conducting a preliminary skimming of the document's contents. This entails a rapid examination to ascertain its structure and identify the chapters deemed relevant for comprehensive analysis. Subsequently, a keyword search is conducted on the document, employing predefined keywords to delineate sections pertinent to the analysis. The relevant text passages contained within these sections are extracted while maintaining a reference to the original page for the purpose of facilitating traceability. Finally, the text passages are assigned matching indicators and categories for subsequent analysis and categorization.

This process has been the main inspiration for the development of the conceptual end-to-

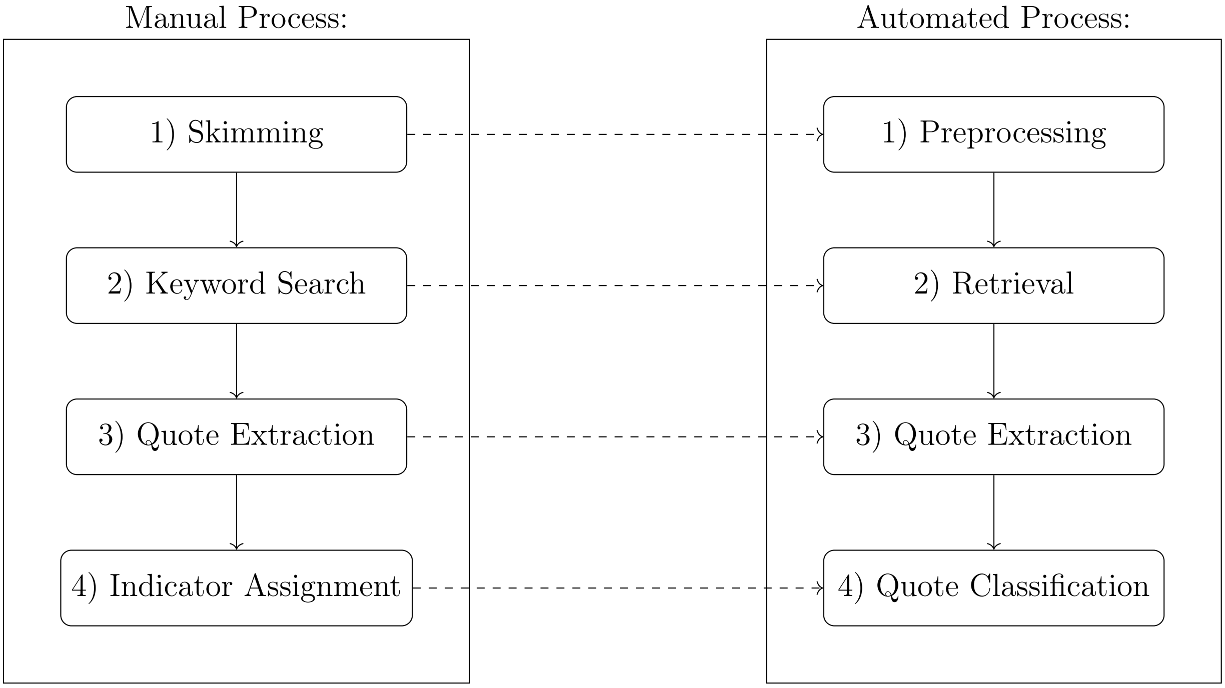


Figure 12: Steps entailed in the manual analytical process of climate policy analysis (*left*) and projection of the manual analytical process to an automated pipeline (*right*).

end pipeline, which tries to mimic the manual process by projecting the individual steps into modules of an automated process.

The initial skimming process translates to a preprocessing module, which accesses the document and extracts raw text accompanied by semantic and syntactic information, such as header structure and content. The text and meta-information is stored in an index that can be accessed by the following modules. The retrieval module translates the keyword search to an automated process of querying the index for relevant textual information, with regard to keywords or queries initially defined by the user. The extracted text undergoes the subsequent step of quote extraction, where the precise sentences and sections, containing the relevant information, are extracted from the text. In the final module, indicators are assigned in a text classification setting, analogous to the indicator assignment by the human analyst. The entire mapping is illustrated in Figure 12.

4.4 Pipeline Architecture

The conceptual pipeline is organized in modules, each presenting a corresponding input and output, which is the intermediate result yielded by the module. The entire proposed pipeline is illustrated in Figure 13 and will be further discussed throughout the remainder of this chapter. Having established the general outline, the subsequent sections examine the individual modules that comprise the end-to-end pipeline, with a particular focus

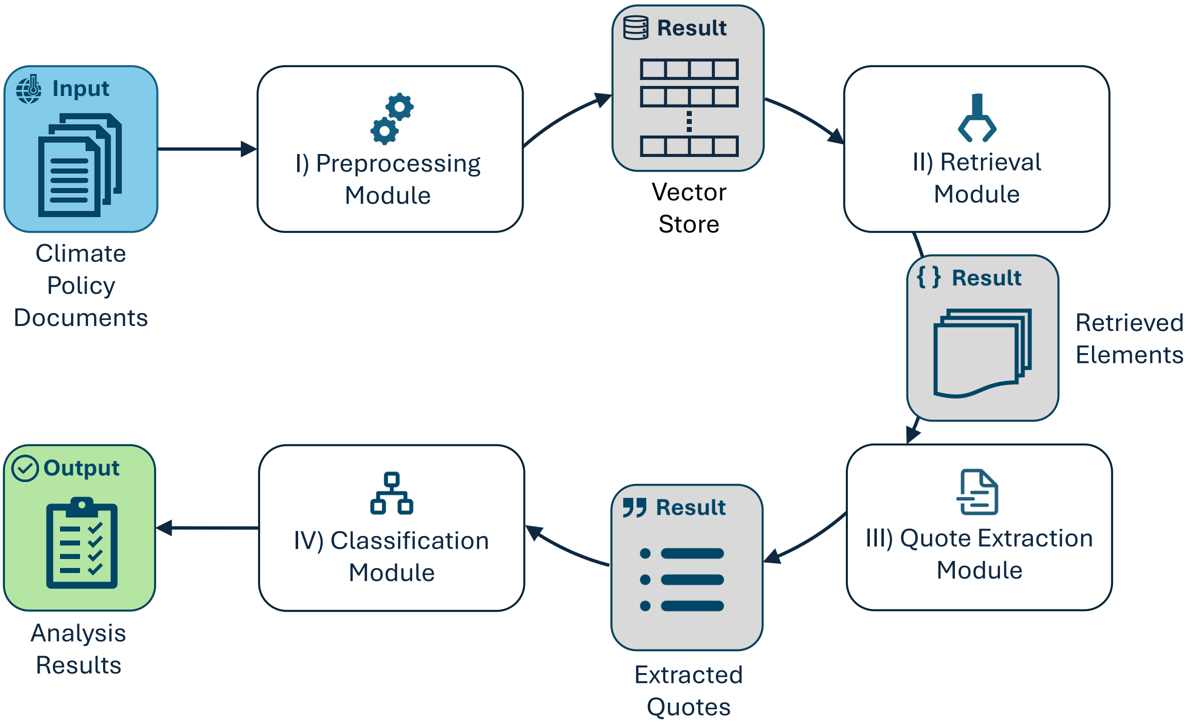


Figure 13: Overview of the entire proposed automated policy analysis pipeline.

on the rationale behind the selected design decisions. Each module description is accompanied by a diagram that illustrates the procedures and methods entailed within the component.

4.4.1 Preprocessing

This section discusses the components that constitute the preprocessing module of the pipeline. In particular, it outlines the key considerations underlying the parsing of documents, summarization, and the indexing process. The entire preprocessing process is illustrated in Figure 14.

Document Parsing

Within the preprocessing pipeline, the document parser is tasked with the separation of the information representations contained within the file and their subsequent extraction. The information contained in the policy documents is organized into three main representation categories: textual elements, tabular elements, and image elements. Given the distinctive characteristics and challenges associated with each of these information representations, they are addressed separately in the subsequent processing steps.

The vast majority of policy documents are published in PDF format. PDF processing for text and syntax extraction poses several challenges due to the inherently complex and variable structure of PDF files. Unlike formats designed for textual data, PDFs are

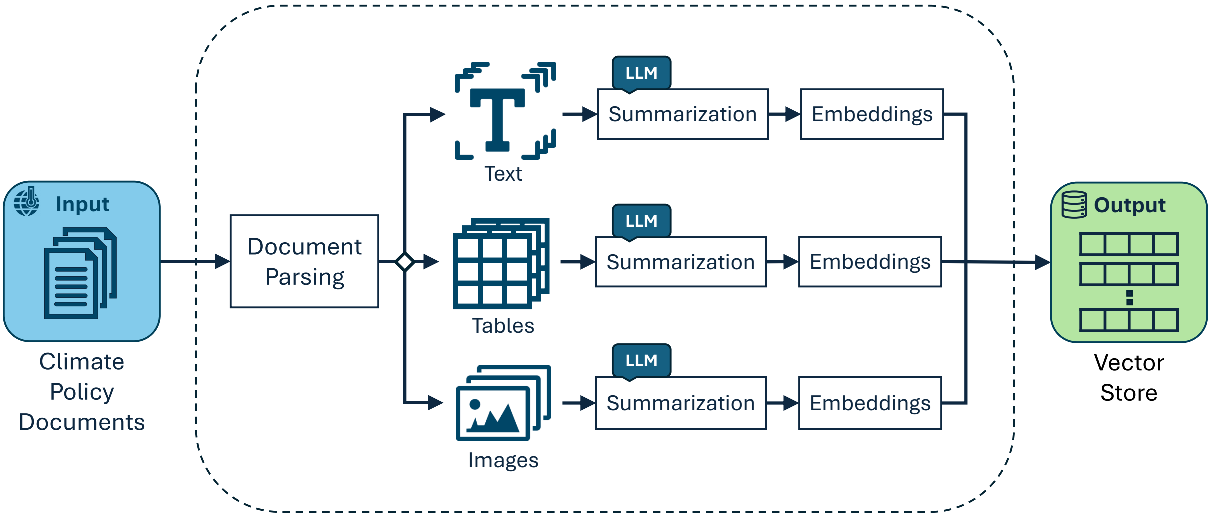


Figure 14: Conceptual illustration of the preprocessing module.

primarily intended for preserving the visual layout of documents for printing purposes. For instance, PDFs lack structural information, making it difficult to discern elements like headers, footers, page numbers, and tables from the main body of text (Hong et al., 2021). Heuristic methods or ML algorithms are therefore required to interpret the layout and structure of the content.

Further complicating the process, the text in PDF documents is typically absolutely positioned, meaning each character's placement on the page is fixed without inherent meaning about its relational context (Hong et al., 2021). This absolute positioning can lead to issues in identifying the logical flow of text, particularly in the presence of multi-column layouts. The presence of additional textual objects, such as running titles, page numbers, or figure captions, introduces an additional layer of complexity to the determination of text flow, potentially leading to confusion with the actual narrative text body.

The challenges in analyzing PDF mentioned above require applying a parser that is able to recognize the inherent structure and contents of PDF documents. Typically, computer vision models are employed for this purpose, which is why this task is commonly referred to as Document Image Analysis (DIA) (Shen et al., 2021)(Huang et al., 2022). For alternative document formats, such as markup language-based formats (HTML) or XML-based formats (DOCX), the parsing process involves less complexity, as the document layout is embedded within the file (Hong et al., 2021).

Summarization

The NLP task of summarization offers a number of practical and theoretical benefits in the context of document analysis. From a practical perspective, summarization of text documents is often a required first step, as the length of the text at hand may exceed the context window of the applied language model. This is frequently the case in RAG applications, where a number of documents is retrieved and appended to the input context. To address this issue, advanced summary strategies have been developed, including *Map-Reduce*, in which separate summaries for each document are created and combined, and *Refine*, in which document summaries are generated incrementally (Chakraborty, 2023).

From a theoretical viewpoint, summarization is a process that distills the essential concepts of a text by filtering out distracting details and retaining the most pertinent information contained in a document (Clark et al., 2023). With regard to the challenges and requirements associated with the analysis process of climate policy documents, it is therefore considered advantageous to incorporate a summarization step on the extracted text elements, with the objective of reducing the overall complexity of the narrative structure and the specific terminology employed.

The same intuition of distilling information is also applied to tables and images. In this case, however, the summary assumes an additional function, as a preliminary information conversion is required in order to process the information from tables and images using language models. For table processing, this step is referred to as serialization, where the structured tabular information is converted into a text representation (Fang et al., 2024). Serialization techniques are broadly classified into text-based, where a table is translated into readable representations such as Markdown, HTML or natural language (Singha et al., 2023)(Hegselmann et al., 2023), and embedding-based, where fine-tuned PLMs are applied to encode tables into embeddings (Herzig et al., 2020). The proposed automated summarization step is a mixed serialization method, as it applies LLM to create descriptions of the tables' content.

The processing of images follows a similar rationale. In this case, however, a multimodal model is required to process the content of the image, which is capable to either obtain the raw image or an encoded representation as input. In the context of climate policy documents, images that contain scanned text are of particular significance, as the documents occasionally contain scanned sections or are fully scanned versions of printed policy documents. In the event that the preceding document parser lacks the capacity to classify scanned sections as text elements using OCR functions, it is necessary to incorporate the summarization step in order to ensure that the information contained in these sections is not lost.

Indexing

Indexing comprises the steps of transforming text into embedded dense vector representations, and storing these vectors in a vector store, which is a database designed to efficiently store vectors of a fixed size and additional metadata related to the vector, including the original text. Indexing and vector stores are key components of RAG applications (P.

Lewis et al., 2021), as they rely on this vector storage option and the additional functionality of querying the vectors to retrieve the necessary information for the answer generation.

The proposed pipeline employs a similar mechanism. The embeddings generated from the summaries of each information representation are stored in a vector store, along with the original information extracted by the parser. In the case of text elements, this refers to the original text as extracted from the document. For tables, the original representation refers to a serialized representation of the table. Images are stored in an encoded format, such as Base64 representations (Josefsson, 2006).

For traceability reasons, additional metadata to be stored in the vector store includes the page number and the source document, from which the information was extracted. Typically, a unique identifier is also attached to the vectors.

4.4.2 Retrieval

The processes comprising the retrieval module are closely related to those performed for the task of IR, as described in Chapter 2.2.4. As in IR, the module is responsible for the task of extracting relevant information from a corpus of unstructured data based on user queries. In this case, the corpus of unstructured data refers to the vector store generated during preprocessing.

As a second input to the module, one or multiple queries or keyword are passed to the module by the user during runtime. These queries or keywords relate to specific information in the documents targeted for retrieval. It is also conceivable that well-working queries or keywords are incorporated in the module configuration beforehand, in the case that targeted information is static and known a priori. For the general conception, it was considered beneficial to model the queries as external input for enhanced flexibility of the pipeline.

The selected queries or keywords are transformed to text embeddings and the resulting dense vector representations are leveraged for vector similarity calculation on the entire data set contained in the vector store. The vectors yielding the highest similarity to the query are retrieved as results. This process is also referred to as *semantic search* (Trautmann, 2023), as the search is based on semantic similarity defined by proximity in vector space (see Chapter 2.2.4).

A final input necessary to the process consists of a parameter to influence the amount of results returned by semantic search. This is achieved either by defining the exact number k of elements to be retrieved from the vector store, resulting in the retrieval of the k nearest neighbors, or by defining a threshold θ for the minimum vector similarity value

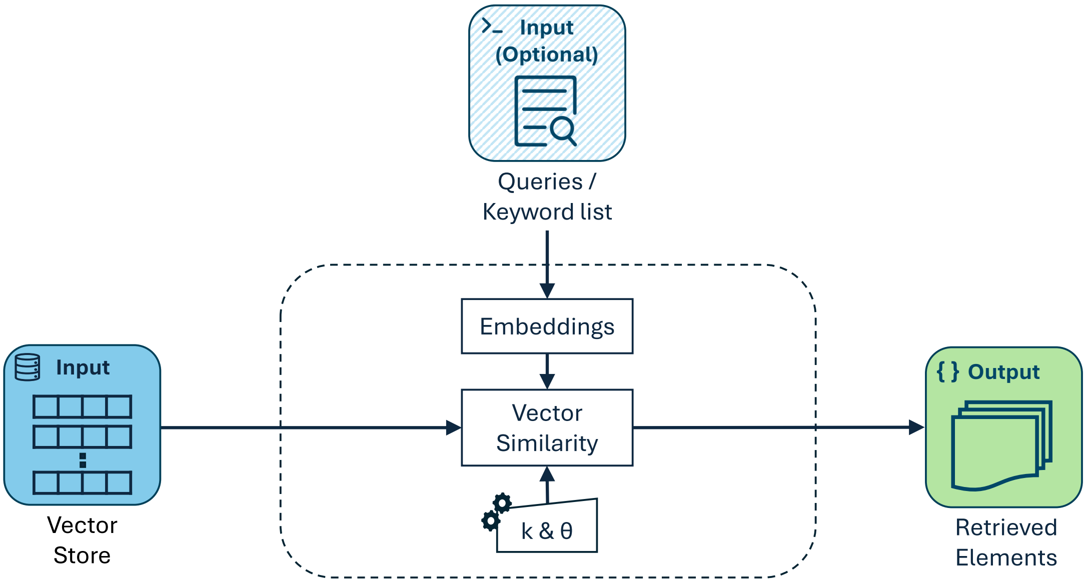


Figure 15: Conceptual illustration of the retrieval module.

required to be considered a vector close to the query.

All retrieved elements are stored in objects that can be utilized for subsequent processing, such as JSON or XML files. In addition to the extracted vectors, these also obtain the entire metadata which is stored in the vector store, such as the original texts or page numbers.

The complete retrieval module is illustrated in Figure 15.

4.4.3 Quote Extraction

The quote extraction module is a simple linear concatenation of three processes, as illustrated in Figure 16. Each of the elements extracted undergoes these processes, starting with the actual quote extraction step. This is performed by prompting an LLM for the extraction of a list of relevant quotes from the original text of the retrieved elements. In the case of images, the created summaries from the preprocessing module (see Chapter 4.4.1) are employed for extraction of quotes. For table elements, the input depends on the applied serialization methods during preprocessing. If the raw text content of the table was extracted during parsing, it is the preferred input to this module.

If the LLM returns a list of quotes during the extraction process, these quotes are subsequently provided as input to a second LLM which is prompted to select relevant quotes for the task at hand from the input list. This prompt chaining step is introduced as the quote extraction may become a complex task depending on the desired information from the document. While the first LLM focuses on extracting candidate quotes from the input text, the second LLM is intended to refine this selection.

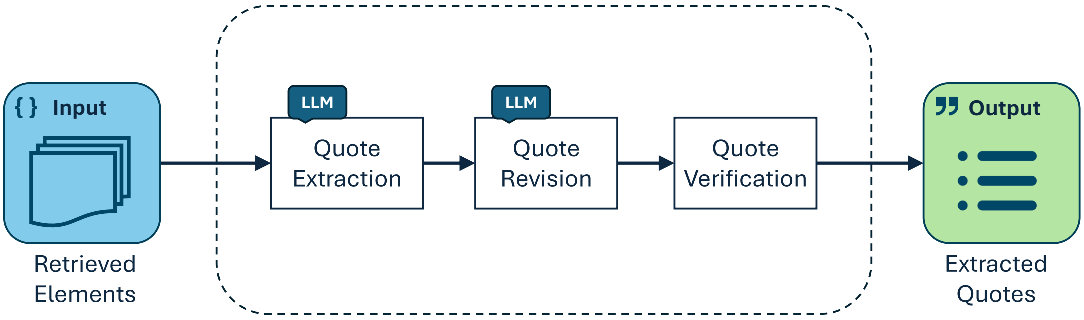


Figure 16: Conceptual illustration of the quote extraction module.

The splitting of the quote extraction step is inspired by a chain prompting technique referred to as *refinement*, which is typically used to improve results in text generation tasks such as summarization (Sun et al., 2024). This technique is motivated by the human writing process, which can be divided into drafting and refinement of the initial draft. In advanced settings, the feedback for refinement of the initial drafts is provided by additional LLMs (Madaan et al., 2023).

The third process involved in the quote extraction module addresses the issue of hallucination, which is a common challenge encountered during conditional text generation with language models. As defined in requirement 9 in Section 4.2, it is a key necessity for the extracted information to be based on the actual content of the policy document. It is thus essential to prevent the extraction of hallucinated quotes. This is achieved by assessing whether the extracted quotes are present in the original text, or the summary, in the case of images.

4.4.4 Quote Classification

Once the quotes are extracted, they undergo the final module, which is tasked with the allocation of classification labels to each quote. The specification of the class labels is dependent on the use case and is assumed to be configured beforehand. In the scope of this conceptual description of the module, only the general subdivision into targets, mitigation measures, and adaptation measures is specified, as shown in Figure 17.

The number of defined categories is potentially vast, depending on the analysis use case. For this reason, a hierarchical classification approach is proposed, where a classification algorithm passes through one or more nodes of a predefined taxonomic hierarchy (Stein et al., 2019). The categories of measures and targets frequently exhibit an implicit division into subcategories or may be merged into higher-level categories, facilitating the task of generating the inherent taxonomy.

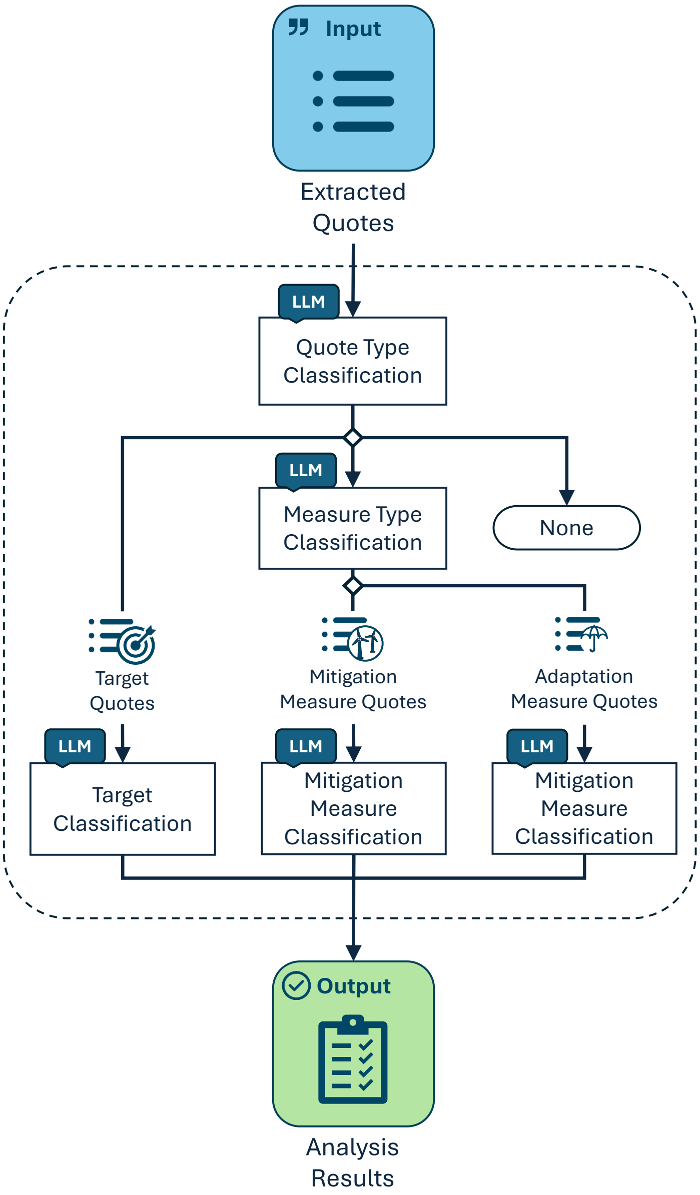

Input
Extracted Quotes
LLM
Quote Type Classification
LLM
Measure Type Classification
None
Target Quotes
Mitigation Measure Quotes
Adaptation Measure Quotes
LLM
Target Classification
LLM
Mitigation Measure Classification
LLM
Mitigation Measure Classification
Output
Analysis Results

Figure 17: Conceptual illustration of the quote classification module.

The task of correctly classifying quotes from climate policy documents into potentially fine-grained categories is the task in the pipeline that requires the highest level of domain-specific knowledge. It is, therefore, the task that presents the greatest challenge to the initially stated hypothesis that a fully automated end-to-end pipeline can effectively operate without domain-adapted or fine-tuned models. The applied prompting method employed to achieve the desired output by the applied general-task LLM thus assumes particular significance. In this particular instance, the selection of a hierarchical classification approach and the partitioning of the categories into subcategories represents a case of task decomposition, a technique to reduce complexity as delineated in Chapter 2.2.3.

To initiate the process, a first multi-label classification is performed to distinguish between quotes referring to targets or measures. If none of both apply, the quote is discarded as irrelevant.

The subsequent step in the classification hierarchy is the subdivision of quotes containing measures into two categories: mitigation measure quotes and adaptation measure quotes. This requires a binary classifier that is capable of distinguishing between the concepts of mitigation and adaptation. According to IPCC (2022), mitigation measures concern interventions to reduce emissions or enhance the sinks of greenhouse gases, whereas adaptation measures include adjustments to actual or expected climate and its effects, in order to moderate harm or exploit beneficial opportunities. The measures stated in the quotes generally serve to reduce emissions solely indirectly, thereby complicating the distinction between emission reductions and climate adaptation.

The same categorization into adaptation and mitigation may also be applied to the quotes classified as targets. However, the vast majority of targets in climate policies are related to mitigation, and therefore the distinction is of less relevance in this case. A more frequent distinction is drawn between conditional and unconditional targets (Pauw et al., 2022), as well as between greenhouse gas-related and non-greenhouse gas-related targets (Wartmann et al., 2023).

As mentioned before, all classification tasks included in the module are intended to be performed by general-purpose LLMs, without specific training. In consequence, no training data is required for this process, eliminating the necessity for the time and resource-consuming processes of data collection, data annotation, and model training. For the classification of fine-granular subdivisions, however, it can be advantageous to provide the model with individual examples to familiarize it with the task and boost their classification performance. The emergent ability of in-context learning inherent of LLMs (see Chapter 2.2.2) is applicable for this purpose, to perform few-shot prompting (Chapter 2.2.3). While the model performance may increase by specifying an individual example, the additional performance increase is observed to be limited after the provision of the second example (W. Chen, 2023). In order to keep the annotation effort low, the zero-shot approach is therefore only enriched with one or two examples in particularly difficult label distinctions.

The classified quotes constitute the terminal output of the automated pipeline. Depending on the use case, these outputs can subsequently be post-processed in various manners. For example, it is conceivable that the quote-label-tuples are extracted into structured text formats, such as CSV or JSON files. These files serve as dataset for further analyses, such as sector or country-specific assessments, or they may be integrated into a dedicated database system, as proposed by Juhasz et al. (2024), Sietsma et al. (2022) or Swarnakar and Modi (2021).

4.5 Additional Considerations

The introduced pipeline is designed in accordance with the requirements detailed in section 4.2, and therefore taking into account the stated challenges related to the task of analyzing climate policy documents described in Section 4.1. The following section delineates additional benefits resulting from the selected design.

As previously indicated, the design offers the significant advantage of relying solely on general-purpose LLMs. This circumvents the time and resource-consuming processes such as data collection, annotation, and training. However, no information has been provided on the selected models, as the pipeline is intended to be model-agnostic. As a result, it should be possible to integrate and exchange models according to the users' preferences, with a particular option being the integration of domain-adapted PLMs in order to increase performance on individual climate-related tasks.

One requirement which has not yet been addressed explicitly is multilingualism, as defined in requirement 3. State-of-the-art LLMs count with inherent multilingual capacities, as they are trained on data in different languages. Therefore, they implicitly learn to perform MT. However, the language capabilities depend upon the degree to which training data is available in the respective language. For this reason, Robinson et al. (2023) draw a distinction between high-resource languages (HRLs) and low-resource languages (LRLs). As a reference point for the subdivision, the authors employ the number of Wikipedia pages in the respective language, which is found to strongly correlate with the language capability of a model. Given that the minimum requirement on language capabilities is set on the six UN languages, which are among the top languages in terms of Wikipedia page numbers [10], it can be reasonably assumed that state-of-the-art LLMs have sufficient capacity in these languages. Extending the pipeline to documents in other languages, in particular to those smaller in terms of resources, would require an additional MT step, to translate the document content into a language in which the applied LLMs are more confident, as proposed by Thulke et al. (2024). This process would ideally be located between document parsing and summarization in the preprocessing module. A detailed study of the application of prompting to the task of machine translation is provided by (Zhang et al., 2023).

An open design question is whether the results are to be translated back into the original language. This could be a particularly useful step for the verification of quotes, as the original text is required to verify the quotes and avoid hallucination. In contrast to the proposed method in section 4.4.3, this setting would require the additional application of advanced methods, such as BiLingual Evaluation Understudy (BLEU), to evaluate the accuracy of translation for the individual extracted quotes.

[10] List of Wikipedias: https://meta.wikimedia.org/wiki/List_of_Wikipedias, retrieved August 15, 2024

5 Implementation

The conceived end-to-end pipeline presented in Chapter 4 has been applied to a specific use case to test its functionality. The following chapter introduces the setting of the use case, the applied data set, as well as the detailed implementation steps to a fully functional automated analysis pipeline prototype.

5.1 Use Case

The use case was developed in cooperation with the GIZ, which is the main German service provider in the field of international cooperation for sustainable development[11]. More specifically, the use case is a component of the project Mobilize Net-Zero (MNZ)[12][13], funded by the International Climate Initiative (IKI) of the German Federal Ministry for Economic Affairs and Climate Action (BMWK).

The main objective of the MNZ project is to support governments in their commitment to the Paris Agreement (see Chapter 2.3.3), specifically in the decarbonization of the transport sector, which accounts for one quarter of energy-related CO2 emissions, and has the fastest growth in CO2 emissions among combustion sectors globally (SLOCAT, 2023). A key activity of MNZ to address this issue is to monitor climate ambition in transport through the analysis of climate policy documents issued by national governments, including NDCs, LTSs, national climate policies and laws, national transport plans and further legal documents.

The analysis of the documents consists of the extraction of targets and measures, which are categorized and assigned with labels from an indicator set elaborated by climate policy experts. The set of indicators is arranged in a taxonomy that provides a thematic subdivision and reflects the relationships and linkages among them, as illustrated in Figure 18. The complete list of indicators and their description is provided in Appendix A.1.

The results of the analyses are recorded in a multi-functional relational database that provides a comprehensive repository of all previous analyses. Further analyses, such as country profiles, are conducted on this basis. The collected database is also published in a free, accessible format via the NDC Transport Tracker[14], thus making them available to other researchers and organizations.

[11] German Development Cooperation (GIZ): https://www.giz.de/en/, retrieved August 15, 2024

[12] Mobilize Net-Zero: https://www.changing-transport.org/project/mobilize-net-zero/, retrieved August 15, 2024

[13] Mobilize Net-Zero: https://www.international-climate-initiative.com/PROJECT1851, retrieved August 15, 2024

[14] NDC Transport Tracker: https://www.changing-transport.org/tracker/, retrieved August 15, 2024

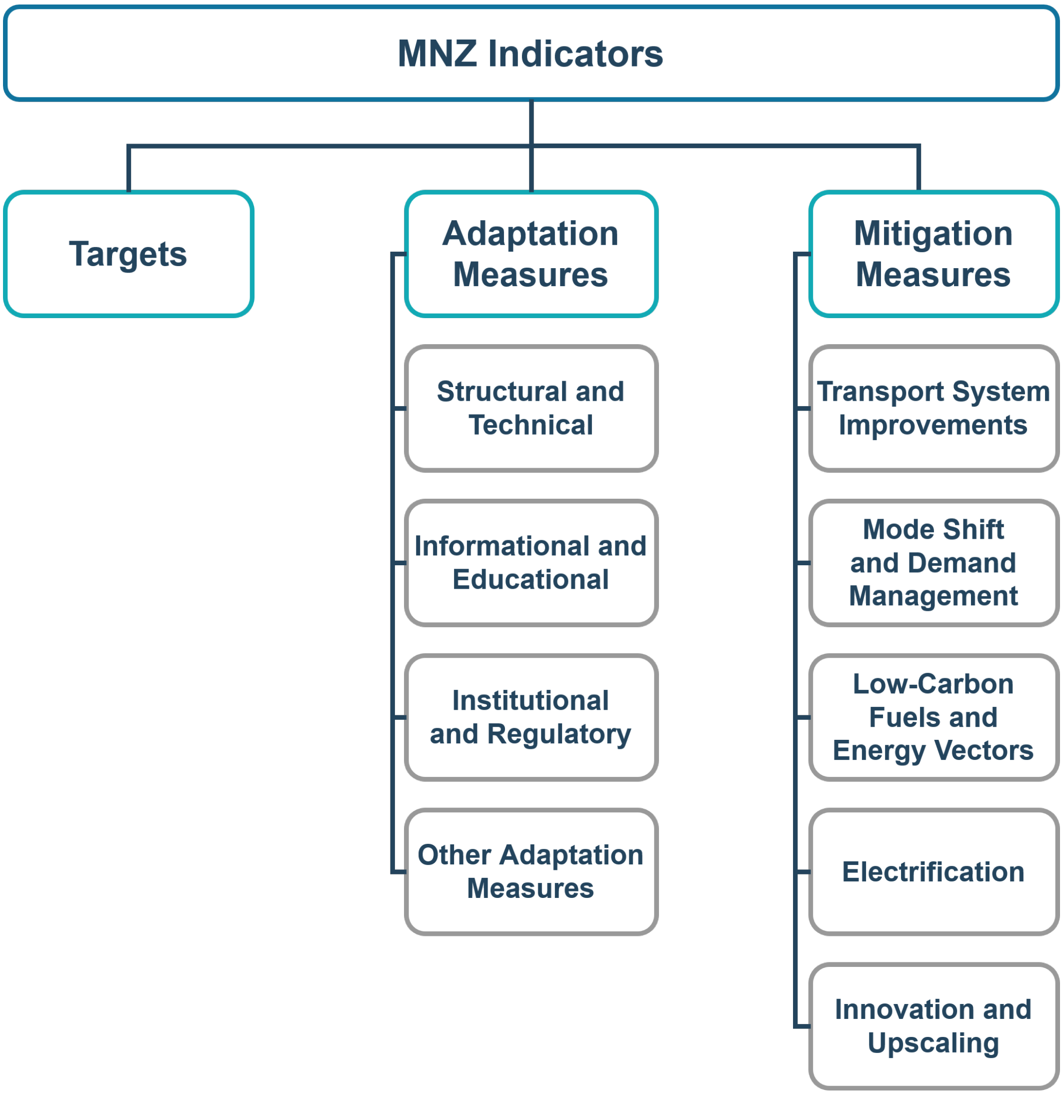

MNZ Indicators
Targets
Adaptation Measures
Structural and Technical
Informational and Educational
Institutional and Regulatory
Other Adaptation Measures
Mitigation Measures
Transport System Improvements
Mode Shift and Demand Management
Low-Carbon Fuels and Energy Vectors
Electrification
Innovation and Upscaling

Figure 18: Taxonomy of indicator labels as applied by Mobilize Net-Zero (MNZ).

.

5.1.1 Use Case Requirements

The implemented pipeline prototype presented in this thesis follows the objective to assist the manual document analysis process for the NDC Transport Tracker within the MNZ project. The described use case imposes a series of specific requirements to the automated pipeline, complementing the general requirements outlined in Chapter 4.2. These additional requirements are briefly described in the following:

1. The analysis is primarily concerned with NDCs and LTSs. These documents are published on official UNFCCC platforms (NDC Registry[15] and LTS Portal[16]), and thus retrieved from these locations.

[15] NDC Registry: https://www.unfccc.int/NDCREG, retrieved August 15, 2024

[16] LTS Portal: https://www.unfccc.int/process/the-paris-agreement/long-term-strategies, retrieved August 15, 2024

2. The predefined indicator set, as described in Appendix A.1, provides the target values for the classification of extracted quotes. The additional taxonomy illustrated in Figure 18 provides a guideline for the hierarchical classification.

3. The indicator labels are non-exclusive, meaning that one quote may be associated to various categories. This results in a multi-label classification task, which must be accounted for in the classification module (see Chapter 4.4.4).

4. For enhanced traceability and to facilitate transparency to the automated process, it is additionally required to include the page number to each classified output. This enables the human analyst to verify and assess the classification applied in the original document, if necessary.

5. In order to facilitate the subsequent processing of the pipeline results, it is preferable to output the results in a format that can be readily imported into a spreadsheet application, such as CSV or Excel. Furthermore, the alignment of the output structure with the database structure simplifies the transfer of results.

5.1.2 Data

The data provided for the elaboration of the use case is the entire data set provided by the NDC Transport Tracker. It consists of the results of the manual analysis of 512 climate policy documents from 196 countries, including all 195 Parties to the Paris Agreement. Of the 512 documents, 405 documents are NDCs, either in their initial submission or an updated submission and 88 are LTSs. The remaining 19 documents comprise national policy documents.

From this entire corpus, a total of 3100 measures and 1122 targets were extracted by the analysts. All of these were annotated with at least one corresponding label from the indicator set (A.1). In addition to the indicators, further data is gathered on the quotes, such as the target year or the A-S-I (Avoid-Shift-Improve) categorization. However, this is not pertinent to the use case under consideration, and thus will not be further addressed.

The entire annotation was performed in MS Excel and the data is provided in such spreadsheet format. The file is freely available on the NDC Tracker Website[17]. The properties of the data set are further presented in Table 7.

[17] NDC Transport Tracker Data Set:
https://www.changing-transport.org/wp-content/uploads/NDC-Transport-Tracker_full-database_040823.xlsx, retrieved August 15, 2024

# Countries:	**196**
# Documents:	**512**
. . . of which NDCs:	405
. . . of which LTSs:	88
. . . of which Other:	19
# Quotes:	**4222**
. . . of which Targets:	1122
. . . of which Measures:	3100

Table 7: NDC Transport Tracker data set description.

5.2 Pipeline Prototype

The listed case-specific requirements, as well as the general requirements to the pipeline concept defined in Chapter 4.2, set the frame for the prototypical implementation of the pipeline concept. The following sections provide a detailed explanation of the methods and steps entailed.

As in the proposed concept, the implementation process follows the guideline of subdividing the pipeline into four modules, each representing a closed and interchangeable component of the application, with predefined input and output objects.

The pipeline is implemented in Python 3.8.15[18], a release that is compatible with the employed packages. The implemented prototype is publicly available on GitHub[19].

5.2.1 Preprocessing

Following the descriptions in Section 4.4.1, the preprocessing module constitutes the initial component of the pipeline, with the task of processing the rare document and store its content in embedded representations in a vector store.

Two options are presented to enter documents as input to the pipeline. If the document has previously been manually retrieved and downloaded from the official platforms (see Section 6), the file can be provided by specifying the path where it is stored in the local repository. Alternatively, it is possible to specify the URL to the document, in which case

[18] Python 3.8.15: https://www.python.org/downloads/release/python-3815/, retrieved August 15, 2024

[19] Pipeline Repository: https://www.github.com/nicolas-becker/climate-policy-miner, retrieved July 23, 2024

the download process is performed by the application and the document is accessed via the downloaded file. The URL is handled via the requests package[20].

Document Parsing

A number of python libraries exists for the essential task of document parsing. This prototype implementation leverages the extensive functionalities provided by Unstructured[21]. At the present time of this thesis, Unstructured library offers both a paid subscription and free API version. The latter offers the core functionality set, but is limited to 1000 pages per month, which is deemed sufficient for the specified use case.

Before the document is parsed, the document type is inferred. Unstructured supports a variety of file types, including PDF, DOCX and HTML, which are relevant for the use case. In order to detect the file type, the file's MIME type is inferred by applying the libmagic file type identification library[22].

Once the file type is inferred, the corresponding partitioning method is applied. This step consists of the extraction of structured content from the raw, unstructured documents and is important to enable the preservation of the semantic structure of the documents. The individual extracted contents, named *elements*, are categorized in different element types, such as titles, tables, narrative text, images, and list items. This is achieved by applying a DIA model, which is a vision model trained on images of documents, with the objective of detecting pieces of information within the document and the layout in which they are organized (Shen et al., 2021).

A particular case of policy documents is the frequent occurrence of scanned document pages. As described in Section 4.2, the processing of the information contained within these pages requires OCR methods. The unstructured library facilitates the application of OCR through the integration of the Tesseract OCR library[23]. Consequently, scanned text is detected and extracted as common text. Tesseract is also applied to detect the language in an element's text.

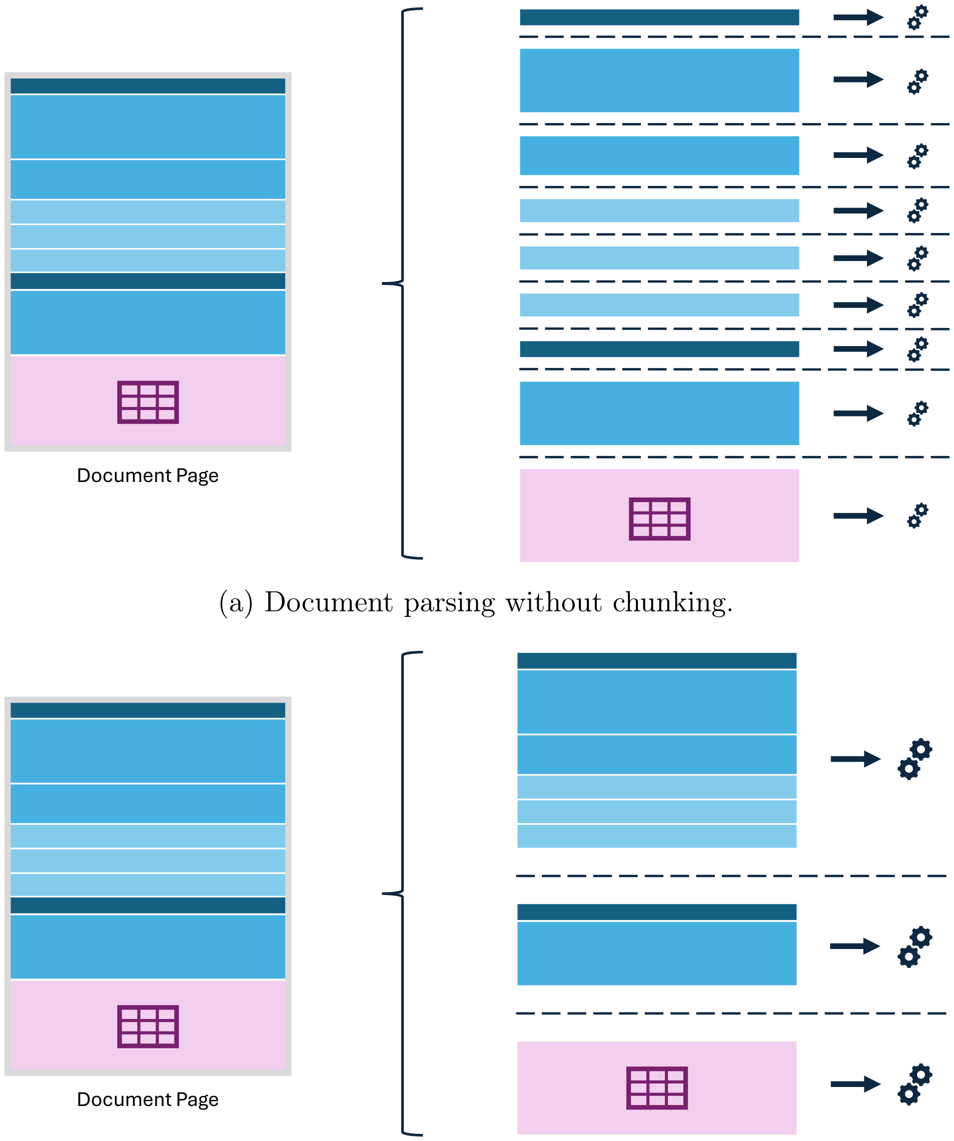
The partitioned document containing its elements and respective element categories is the basis of subsequent analysis steps. Before proceeding in the pipeline, an optional pipeline configuration includes an intermediate step consisting of *chunking*, which is the process of merging individual elements into larger elements. This has the benefit of expanding the context of information contained within the individual elements, and reducing the amount of elements to be processed in the subsequent steps. Based on the document layout inferred with the DIA-model, chunking is performed based on the document structure

[20] Requests library: https://www.requests.readthedocs.io, retrieved July 23, 2024

[21] Unstructured.io: https://www.github.com/Unstructured-IO/unstructured, retrieved July 23, 2024

[22] Libmagic library: https://www.github.com/ahupp/python-magic, retrieved July 23, 2024

[23] Tesseract library: https://www.github.com/tesseract-ocr/tesseract, retrieved July 23, 2024



(a) Document parsing without chunking.

(b) Document parsing with chunking.

Figure 19: Difference between document parsing without chunking (*top*) and with chunking (*bottom*). Chunking reduces the number of extracted entities and preserves syntactic and contextual information between elements.

.

and element type. For instance, textual elements, such as narrative text or item lists, are only merged, if they are subordinated to the same title element. Tables and images are not included in the chunking procedure and extracted separately. Figure 19 illustrates this process. The configuration chosen for the chunking is specified in Table 8.

The extracted elements represent the output of the document parsing. They are stored in Pydantic[24] objects, along with metadata that is gathered during extraction. This entails the page number, the language, and the coordinates of the extracted text within the PDF

[24] Pydantic library: https://www.github.com/pydantic/pydantic, retrieved July 23, 2024

Chunking Parameters	
`min_length`	10
`max_length`	8192

Table 8: Chunking configuration. `min_length` refers to the minimum number of characters required for an element to become a stand-alone chunk. Elements containing fewer characters are always merged with the subsequent chunk. `max_length` refers to the maximum allowed character count of chunks. Chunks reaching this character count are no longer populated with elements, and a new chunk is created.

document. The unstructured library is thus capable of handling the challenges regarding PDF parsing described in Chapter 4.4.1.

For the table elements extracted from the document, the parser additionally extracts an HTML representation of the table, as an alternative serialization method to the plain text extracted (see Chapter 4.4.1), as inferred by the Tesseract library. The tables images extracted are provided in Base64 encoding (see Chapter 4.4.1).

Summarization

Once the tables, images and (chunked) text elements are extracted, they undergo the summarization step, which is performed by prompting an OpenAI GPT-3.5-Turbo model. The LLM obtains the original content of the element and is prompted to provide a concise summary of the content. The prompts employed for this task are provided in Appendix **??**. The summarization of images requires a multi-modal model.

As the summarization of images is crucial to process the information contained within the visual representations such as graphics and charts, the summary of images is required to be precise and detailed. The prompt is therefore designed to not only summarize the image in a detailed description, but to categorize the information contained, and, most importantly, extract the text contained. The prompt is displayed in Appendix 26. An OpenAI GPT-4 model was employed for this task, and the Base64 encoded image provided by the parser is conveyed as input to the model.

Indexing

As described in Chapter 4.4.1, indexing is performed though a vector store database, that allows for efficient storage and retrieval of the extracted elements. The vector store is built with Pinecone[25]. For each document ingested, a separate namespace is created within the

[25] Pinecone: https://docs.pinecone.io/home, retrieved July 23, 2024

database. This allows the documents to be queried individually.

Before being ingested to the storage, the elements' summaries are converted to dense word embedding vectors via the application of OpenAI's text-embedding-ada-002 embedding model (OpenAI, 2022), which creates dense vectors of 1536 dimensions. The embedded vectors are the reference values for the queries on the database. All additional information on the element, such as the original content and the metadata retrieved during document parsing, is conveyed to the vector store as metadata.

5.2.2 Retrieval

The vector storage created as output to the preprocessing module serves as input for the retrieval module, as illustrated in the concept description (see Chapter 4.4.2).

The query procedure is conducted with Pinecone, which implements the cosine similarity measure (see Chapter 2.2.4) to calculate the vector space proximity between the indexed document element summaries and the query. These are embedded using the same text-embedding-ada-002 embedding model, to ensure consistent representations between query and index.

In contrast to the conceptual retrieval module, the implemented prototype does not require the queries to be externally specified during runtime. Instead, the selected queries are configured in the pipeline after a conducted assessment. Initially, the query consisted of a keyword list with 8 keywords specifically targeting the indicator set to be extracted. The keyword list is specified in Appendix Table 24. These keywords were subsequently queried and the resulting elements were appended, taking into account possible duplications.

This semantic keyword search approach exhibited two significant shortcomings. Primarily, the repetitive querying resulted in a notable reduction in runtime efficiency. Secondly, the individual keywords were maintained in a highly generalized form, which led to an inability to reliably extract specific information. To illustrate, the keyword "Transport" demonstrated a high degree of similarity with titles or captions such as "Transport measures". However, in instances where the chunking did not align with the title and content, only the title was returned, rather than the contents of interest itself.

To approach this, the semantic keyword search was replaced by a single query, which was formed as a result of an empirical analysis of the quotes contained in the data set. For this purpose, the entire set of quotes contained in the data was merged and tokenized with OpenAI's Tiktoken library[26], to imitate the tokenization occurring within the embedding models. The token distribution was inferred within this set of tokens, in order

[26] Tiktoken library: https://github.com/openai/tiktoken, retrieved July 23, 2024

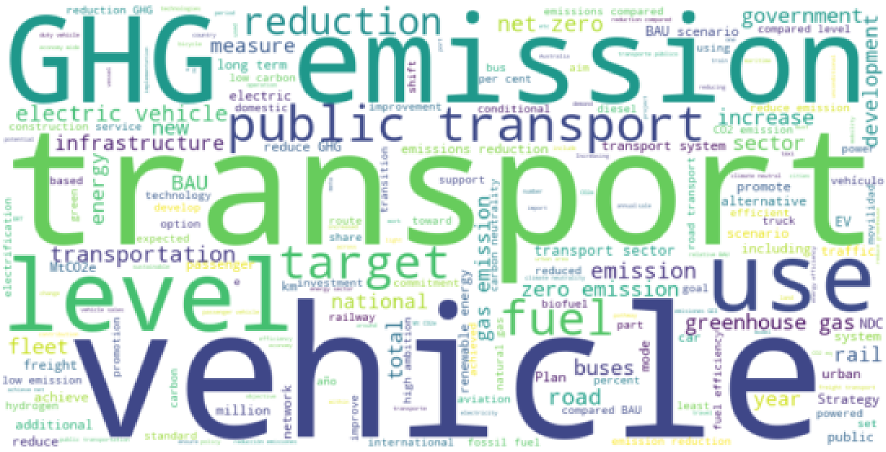


Figure 20: Word cloud of frequent words in targeted quotes.

to identify those tokens appearing frequently in extracted quotes. Previously, stop word and punctuation were removed. This process was analogously performed for the entire text corpus of 16 NDC and LTS documents, and the difference of the two distributions was calculated to correct for tokens occurring frequently within the documents in general. The resulting distribution is illustrated in Appendix Table 32 and provided the basis for a query containing tokens that are frequently contained in the targeted document contents. The tokens that make up 50% of the distribution mass were selected for the query. Figure 20 shows a word cloud created analogously but without tokenization, for illustrative purposes.

The retrieved vectors are returned according to the defined configuration of the input parameters k and θ in the module. Three different configurations have been tested, as indicated in Table 9. The static-k configuration was performed by returning $k = 10$ elements per query. This heuristic method is based on the average number of quotes contained in a document, according to the data set, as described in Table 7. The major drawback of this approach is that it assumes that all documents contain approximately the same number of citations, which is not the case due to the large variance in document length in NDCs and LTSs.

An adaptation to this approach is denoted as dynamic-k, as it adapts the amount of retrieved values to the number of retrieved elements. The problem with this approach is its scaling behavior, as the number of retrieved elements increases rapidly with document size, affecting the runtime of all subsequent pipeline steps. To ensure that only the relevant elements are retrieved even for documents of considerable length, the third configuration is introduced. Here, the θ parameter is additionally employed, which indicates a threshold for the minimum level of similarity required for an entry in the index to be retrieved. The value of 0.75 for this parameter has been identified as the result of several test runs.

Retrieval Configurations	k	θ
Static-k	10	0.00
Dynamic-k	#Elements/3	0.00
θ-Threshold	#Elements	0.75

Table 9: Tested retrieval configurations.

5.2.3 Quote Extraction

The core functionality of quote extraction module consists again in the implementation of prompting techniques with LLMs. The original contents of the elements retrieved from the vector store are passed as input to a GPT-3.5 model, which is prompted to extract quotes from the text that define targets, measures, or actions undertaken or planned. If the text does not contain any of this information, the model is asked to return an empty answer. The output of the model is parsed as a Pydantic object with a single attribute consisting of a list of the extracted quotes. The respective prompt is presented in Appendix Figure 27.

While the original text serves as the basis for the processing of text elements, this is less evident in the case of table elements. As outlined in Section 5.2.1, the document parsing produces two distinct serializations for the extracted tables: the plain text and the inferred HTML representation. Despite the absence of structural information about the tables in the plain text, it is more conducive to the task of quote extraction than the HTML representation. In addition to the inherent issues with the error-prone process of translating tables to correct HTML representations, the inserted HTML tags and formatting alter the original text, making it more challenging to extract verbatim quotations. In the case of image elements, the image description provided in the summary (see Chapter 5.2.1) is utilized as the source for the extraction of the targeted information.

The second step comprising the quote extraction module is the quote revision. As described in Chapter 4.4.3, this step is required to further specify the required information and therefore reduce the number of extracted quotes to limit the amount of quotes to be processed by subsequent steps. The associated prompt is provided in Appendix Figure 28. The output of the model is parsed as a Pydantic object identical to the one produced during the initial extraction.

The final quote verification step is implemented without the application of LLMs as the task consists of searching the extracted quote within the original content to examine the possibility of hallucination. If the quote is not found within the content, it is considered an unverified quote and marked as such in the output. The output of the module consists of a dictionary object containing the entire element information and metadata gathered to this point. It is additionally exported as a JSON file for the purpose of traceability of the process.

5.2.4 Quote Classification

The concluding component of the pipeline consists in the classification module. Within this process, the extracted quotes contained in the output dictionary yielded by the previous module undergo a hierarchical classification process as described in Chapter 4.4.4. The classifiers utilized for this end are, once again, LLMs, specifically GPT-3.5 Turbo.

To guarantee that the models yield the desired class prediction, a tagging technique[27] is employed, which is designed for labelling of documents. To this end, a Pydantic model ("ResultObject") is defined for each classification stage, which receives the potential classification alternatives as attributes and delineates the instructions that the model is required to follow in order to structure the output. The attributes are accompanied by a description that is designed to assist the model in its decision-making process. This is exemplified in Table 10, presenting the design of the ResultObject for the first classification task in the hierarchy, which consists in defining whether the quote describes a target or a measure.

The tagging approach additionally follows the principle of CoT or chain prompting (Chapter 2.2.3), to decompose the overall classification task into subtasks. This serves the objective of reducing complexity for the classifier, as it is prompted to classify each attribute separately. Subsequently, the results for the individual attributes are aggregated to yield an overall result for the classification task.

The example illustrated in Table 10 contains a third column that comprises a set of possible values for each attribute, in this instance a binary *True*/*False* decision. This is due to the requirement that each classification decision is a multi-label classification (see Item 3 in Section 5.1.1), whereby several classes may apply to a single quote. With reference to the example provided, it is thus possible for a quote to describe both a target and a measure (*True*, *True*). Conversely, it is also possible that neither option applies (*False*, *False*), indicating that the quote contains neither target nor measure and is therefore excluded from further classification steps.

In instances where classification decisions prove to be particularly challenging for a general-purpose LLM, individual examples are incorporated into the attribute description. This few-shot approach is intended to further support the model in understanding the attribute, and therefore enhancing its ability to accurately classify the quote. As argued in Chapter

[27] Tagging: https://python.langchain.com/v0.1/docs/use_cases/tagging/, retrieved July 31, 2024

Field	Description	Values
`target`	In an NDC, a 'target' is defined as a quantifiable declaration of intent, with a proposed target year. Does the text refer to a 'target'?	`True`, `False`
`measure`	In an NDC, a 'measure' is defined as an action that is planned to be undertaken, without necessarily stating a target year for completion. Does the text refer to a measure in the transport sector?	`True`, `False`

Table 10: Example of ResultObject: `QuoteTypeObject` Pydantic Model

4.4.4, a single example has shown to improve the comprehension of LLMs in reasoning tasks.

Further steps in the classification hierarchy are guided by the use case-specific taxonomy employed by MNZ, illustrated in Figure 18. In the case of targets, only one classification step follows, which defines the features of the target. The corresponding ResultObject is provided in Appendix Table 25. Measures are further divided between adaptation measures and mitigation measures. As the mitigation measures comprise 82 indicators, the corresponding categories of the taxonomy are employed to specify additional classification steps. The 16 indicators for adaptation measures are classified in a single classification step.

It should be noted that, according to the proposed tagging approach, a fixed prompt can be applied for each classification step. The prompt implemented within the prototype is outlined in Appendix 30. The only adaptation required is to the ResultObject, which is adapted to the specific case.

In accordance with requirement 5 stated in Section 5.1.1, the readily classified quotes are transformed to a structured representation and exported as CSV or Excel spreadsheet. In addition to the extracted quote and the corresponding list of matching labels, the page in the document where the information is contained is also indicated for reference. An additional output of the pipeline is a highlighted version of the input document, containing color-coded highlighting of the extracted quotes within the document, which were created with the PyMuPDF libary[28]. This is intended to serve as a visual aid for the analyst, providing guidance on the relevant excerpts in the policy document.

[28] PyMuPDF library: https://pymupdf.readthedocs.io, retrieved August 04, 2024

5.2.5 Model Selection

Within the implemented pipeline in the scope of this thesis, all tasks involving the application of LLMs were performed with OpenAI models, deployed via a Microsoft Azure instance provided by GIZ.

In addition to the associated convenience, further reasons exist for selecting these LLMs. The GPT model series demonstrates high performance in a number of general benchmark datasets[29] and is also competitive in tasks that are particularly relevant for the use case, such as table analysis (Fang et al., 2024).

While GPT-3.5 Turbo is not the most high-performing model within the GPT series, it offers notable advantages in terms of efficiency. Compared to the larger GPT-4 models, it is both faster and less resource-intensive, making it a cost-effective choice in production environments where maintaining low operational costs is a priority.

All calls of the GPT model were executed via the LangChain[30] framework, which builds upon the concept of prompt chaining, including further components required for NLP applications. It implements wrappers for the OpenAI API and other packages employed for the pipeline, such as Pinecone and Pydantic.

[29] LLM Leaderboard: https://artificialanalysis.ai/leaderboards/models, retrieved July 31, 2024

[30] LangChain: https://www.langchain.com/, retrieved July 31, 2024

6 Evaluation

The implemented pipeline described in Chapter 5 was subducted to a comprehensive evaluation. In order to determine the effect of each individual pipeline component, the components were analyzed separately. The order of modules is reversed in this chapter, with the final task of quote classification addressed first, and the initial preprocessing module addressed last. The following chapter provides a detailed description of the evaluation procedure and the gathered results.

6.1 Classification Module

The first module to be evaluated was the classification module. As the task performed by this module is to assign correct labels to the readily extracted quotes from climate policy documents, the evaluation of this step consists in the calculation of general classification performance metrics generally applied for classification models in ML as described in Chapter 2.1.4. In order to facilitate the assessment of the LLM's performance for each individual classification task, the tasks were performed independently and in isolation. The test data was obtained from the data set provided by MNZ (see Chapter 5.1.2), from which a number of samples were selected, according to the evaluated classification step.

The initial classification step in the described hierarchy consists of the differentiation between targets and measures. A stratified sample was collected, comprising 50 quotes associated to targets and 50 quotes associated to measures, resulting in a total sample of $n = 100$. The zero-shot tagging prompt implemented as described in Chapter 5.2.4 led to a satisfactory F_1-Score of 81.72% on this sample, with precision and recall yielding equally high results, as depicted in Table 11.

The target classification proved to be a considerably more challenging task, necessitating the correct classification of nine attributes (see Table 25) instead of the two quote type options. The test data for target classification consisted of a stratified sample comprising ten examples per target indicator (see Table 21), resulting in a total sample of $n = 100$. The performance on this sample yielded an F_1-Score of 45.36% for the overall performance, with precision scoring slightly higher at 58.47% (see Table 12). However, when disaggregating the results by attribute, it becomes evident that conditionality is the

Quote Type Classification	**Precision (%)**	**Recall (%)**	**F_1-Score (%)**
Zero-Shot Tagging	82.95	80.67	81.72

Table 11: Performance of zero-shot tagging approach for quote type classification .

Target Classification	**Precision (%)**	**Recall (%)**	**F_1-Score (%)**
Zero-Shot Tagging	58.47	45.95	45.36
Zero-Shot Single Prompt	37.79	39.56	37.62

Table 12: Performance of zero-shot tagging approach for target classification. Additional comparison to single prompt approach.

primary factor contributing to the overall weakness of the performance, as the remaining attributes exhibited F_1-Scores ranging from 70.39% to 95.92%, as depicted in Table 13.

However, the efficacy of the tagging approach becomes evident when the results are compared to those of a simple prompting approach, as shown in Table 12. In contrast to tagging, where a task decomposition is performed (see Chapter 5.2.4), and the outputs are transferred to the desired object schema, this is not the case with simple prompting. In this approach, the model was asked to perform the same target classification tasks within a single prompt, by provision of labels and their descriptions. The respective prompt is listed in Appendix 31.

The classification branch of the quotes containing measures is more complex due to the high number of indicators defined for the individual measures (see Appendix Tables 23 and 22). In this case, the balanced stratified training data set contains three examples per indicator, resulting in a sample of $n = 294$ for a total of 98 indicators.

The initial classification step of differentiation between mitigation and adaptation measures yielded a satisfactory F_1-Score of 95.72% on the test set, as depicted in Table 14. The classifier reaches an even higher precision score of 100%, indicating that it is capable

Attribute	**Precision (%)**	**Recall (%)**	**F_1-Score (%)**
Net-Zero	96.86	95.60	95.92
Mitigation/Adaptation	95.35	91.59	93.38
Sector (Transport, Energy, Economy-wide)	89.00	77.78	82.33
Greenhouse Gas (GHG)	66.53	74.73	70.39
Conditionality	43.72	47.37	38.97

Table 13: Quote type classification performance of zero-shot tagging approach. The performance on Conditionality is significantly lower compared to the other target attributes.

Measure Classification	**Precision (%)**	**Recall (%)**	**F_1-Score (%)**
Measure Type	100.00	91.78	95.72
Adaptation Measures	85.19	62.96	70.13
Mitigation Measure	53.27	16.82	23.86

Table 14: Performance of zero-shot tagging approach for measure classification.

.

of reliably distinguishing between adaptation and mitigation measures. The model is also capable of classifying the adaptation measures with an F_1-Score of 70.13%. As illustrated in Table 14, precision again exceeds recall in this instance. The results for the mitigation measures are significantly weaker. This is evident from the F_1-Score, which reaches only 23.86%, as depicted in Table 14. Recall is even lower at 16.82%, indicating that 83.18% of the mitigation measure quotes are either omitted or misclassified.

It is insightful to additionally examine the contrast between the applied hierarchical classification methodology and the classification of measures in a single comprehensive tagging task. This comparison is presented in Table 15, which reveals that all three metrics perform significantly weaker.

In order to facilitate comprehension of the weak outcome for the mitigation measures, an analysis of the results for each individual label is presented in Appendix Tables 26 and 27. The findings reveal that the performance on the individual indicators exhibits considerable variability, and suggest that the indicators are subject to varying degrees of difficulty in their identification. Hence, in instances where the F_1-Score does not exceed 50%, supplementary assistance was provided through the incorporation of illustrative examples from the data set within the definition. In other words, few-shot prompting was employed in the place of zero-shot prompting (see Chapter 5.2.4). The resulting performance gains are provided in Table 16.

Measure Classification	**Precision (%)**	**Recall (%)**	**F_1-Score (%)**
Hierarchical Tagging (avg)	69.23	39.89	46.99
Comprehensive Tagging	33.22	13.87	18.17

Table 15: Performance comparison between hierarchical classification and single comprehensive measure classification.

Mitigation Measure Classification	**Precision (%)**	**Recall (%)**	**F_1-Score (%)**
Zero-Shot Tagging	53.27	16.82	23.86
Few-Shot Tagging	82.07	51.03	60.17

Table 16: Performance comparison between zero-shot tagging and few-shot tagging for mitigation measure classification.

6.2 Quote Extraction Module

With the classification module attested capable of correctly classifying quotes from climate policy documents, the subsequent question is whether the implemented pipeline is able to extract quotes from such documents that contain the necessary information for classification. This task is performed by the Quote extraction module, as described in Chapter 5.2.3.

In order to evaluate this module, a sample of $n = 100$ quotes was drawn from the data set, and the context for each of them was extracted from the source document. The context length was determined with $k = 400$ in order to obtain approximately 1000 characters per text excerpt, taking into account the length of the quotes themselves. This process was successful for 40 of the sampled quotes. The remaining examples were then augmented with 10 negative examples, namely text excerpts that contain no relevant targets or measures. This results in a total test data set of $n = 50$ entries.

The test data was employed to assess the LLM prompting approach designed to extract relevant quotations from text passages. This is a conditional task, but the model output is not predetermined by a set of possible outputs, as it is the case with the labels in the classification task. Thus, the performance metrics applied for the assessment of classification results are not directly applicable in this case. Instead, an increasingly popular practice to evaluate these tasks to evaluate LLMs outputs is by using another LLM as a judge model (Thakur et al., 2024), which is the approach selected in this case.

The LLM-judge is prompted to evaluate the performance of the LLM extracting quotes from the provided text excerpts from the test data set, by comparing the output with a reference output containing the initial quote. The respective prompt is provided in Appendix Figure 29. The evaluation was performed with the LangSmith[31] framework by LangChain, which facilitates the application of LLM judges in an evaluation environment. The experiment was conducted twice, with the judge yielding a correct classification on an average of 89% of the test examples, which corresponds to the recall metric. This

[31] LangSmith: https://smith.langchain.com/, retrieved August 07, 2024

Quote Extraction	**Precision (%)**	**Recall (%)**	F_1**-Score (%)**
Without Revision	96.00	16.73	28.49
With Revision	90.00	21.63	34.87

Table 17: Performance of quote extraction with and without revision.

result indicates that the expected quote was contained in the output for 89 of the 100 tested model extraction runs.

It is important to note that the quote extraction process also extracted a significant number of quotes that were not specified as references. In a third run, 48 of the 50 quotes were correctly extracted, representing a recall rate of 96%. However, a total of 287 quotes are extracted, indicating that a considerable amount of potentially irrelevant information was contained in the output of the process, resulting in a precision value of only 16.73%.

For this reason, a second process of quote revision was introduced to further process the results of the extraction, as described in Chapter 3. In the above example, the number of output quotes was reduced to 208 by chaining the revision prompt. While three correct quotes were also incorrectly removed, the recall remains at a satisfactory value of 90%. In return, the precision score was increased by 5 percentage points to a value of 21.63%, leading to an increase in F_1-Score from 28.49% to 34.87%. In absolute terms, this corresponds to a reduction of 79 quotes in the output, thus reducing runtime and costs in the subsequent classification procedure. This is summarized in Table 17.

6.3 Retrieval Module

It is necessary to evaluate the preceding retrieval step in order to ascertain the extent to which the pipeline is capable of providing the quote extraction module with pertinent sections of the document.

To this end, an additional binary variable, designated "relevance", was incorporated into the metadata of the indexed elements within the vector store. This variable serves to indicate whether the corresponding section of the document contains quotes deemed relevant (*relevance = True*) or not *(relevance = False*). This procedure was performed for a random sample of 15 documents, comprising a total of 120 quotes in the data set.

The test data thus obtained formed the basis for the parameterization of the queries as described in Chapter 5.2.2. To this end, elements were retrieved from the test data set under different parameter settings and evaluated in terms of the extent to which the elements marked as relevant are contained in the results. The results recorded in Table

18 were obtained for the selected θ-Threshold configuration (see Table 9). The search was limited to the value range $\theta \in [0.70, 0.80]$, as the calculated cosine similarity was observed to fluctuate within this range.

The recall value is of particular importance in this phase of the pipeline, as it indicates the proportion of relevant elements in the vector store that is taken into account in the subsequent analysis. Consequently, elements that remain undetected in this step remain so for the remainder of the analysis process, resulting in the loss of pertinent information. In contrast, a lower precision value is less problematic. This metric indicates the proportion of relevant elements among the retrieved elements. A higher value indicates a lower proportion of irrelevant information included in the subsequent analysis step, which has a negative impact on runtime and costs. However, the quote extraction module, and in particular, its quote revision step, was designed to keep the proportion of superfluous information low (see Section 6.2). This constitutes an additional filtering mechanism intended to reduce costs and runtime.

Thus, when selecting θ, it is important to note that the recall should be sufficiently high in order to reduce the loss of information. The visualization of the results in Figure 21 reveals a notable decline in recall at $\theta = 0.75$, while precision exhibits a relatively modest

θ-Threshold	**Precision (%)**	**Recall (%)**	**F_1-Score (%)**
0.70	5.21	96.96	9.89
0.71	5.55	96.96	10.57
0.72	6.35	95.45	11.92
0.73	7.70	95.45	14.25
0.74	9.10	92.42	16.57
0.75	10.60	87.87	18.92
0.76	11.13	71.21	19.26
0.77	14.13	60.60	22.92
0.78	18.64	50.00	27.16
0.79	22.91	33.33	27.16
0.80	34.04	24.24	28.31

Table 18: Element retrieval performance for different selections of θ.

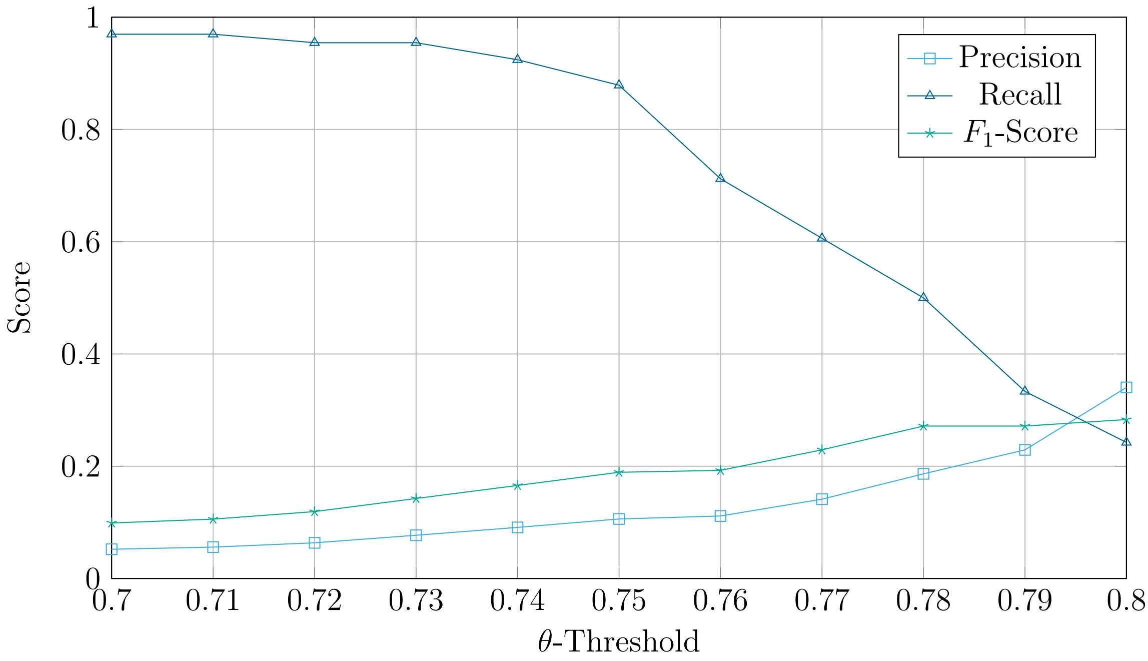


Figure 21: Visualized element retrieval performance for different selections of θ.

.

increase. Accordingly, the composite F_1-Score approaches a slight plateau at this point. The value of $\theta = 0.75$ was consequently selected for implementation in the pipeline, as previously indicated in Chapter 5.2.3.

6.4 Preprocessing Module

Given that preprocessing affects the outcomes of subsequent modules but does not produce any quantifiable output itself, this step was not evaluated independently. Instead, modifications were implemented throughout the pipeline development process, with the expectation that they would lead to enhancements in the subsequent steps.

For instance, the chunking step was incorporated as previously outlined in Section 5.2.1. The advantages of the extended context and structural information, as described, primarily benefit the retrieval process, as larger elements are more likely to contain pertinent information. Additionally, the extended context may potentially enhance classification performance, although preliminary results indicate a potential distraction effect, whereby classification may lose focus on the relevant quote within the surrounding context.

Moreover, the possibility of omitting the summary step for textual and tabular elements was considered, given that it is not indispensable despite the advantages outlined in Section 4.4.1. However, this step is particularly time-consuming and costly, as it represents the only LLM call that is performed on the entire document. Given that the chunking length is tailored to the context window of the embedding model used (see Table 4), the

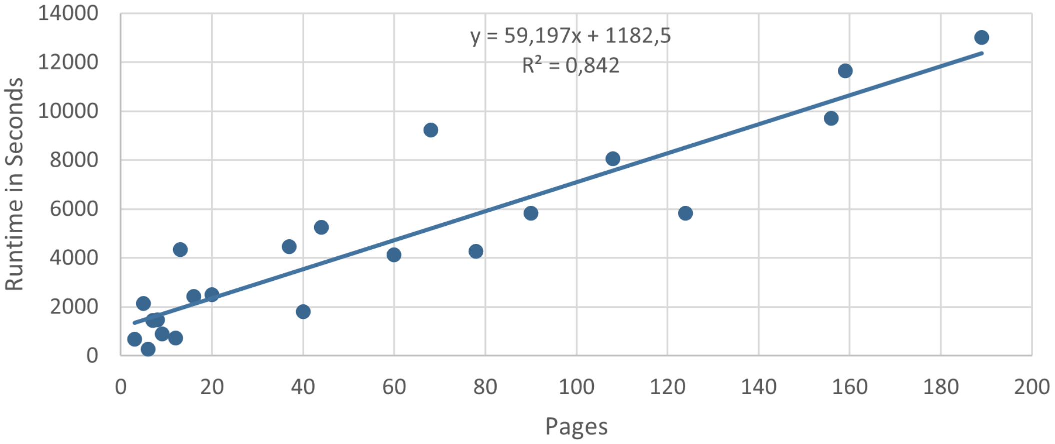


Figure 22: Observed runtime (in seconds) for the entire pipeline on varying document lengths

.

summary is also not necessary from a technical standpoint. The fact that the query is now compared with the original content represented in vector space during retrieval is also expected to enhance the results in this module.

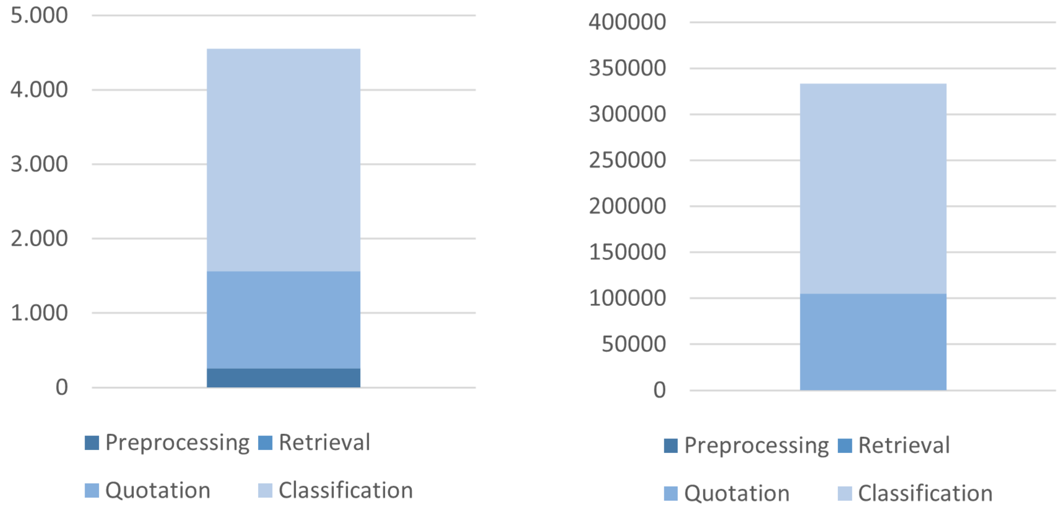
The preprocessing process additionally influences the extraction of quotes, particularly in relation to the selection of table serialization method. In this instance, the HTML serialization approach was superseded by the use of plain text serialization, as detailed in Section 5.2.3.

6.5 Pipeline

In order to examine the behavior of the entire pipeline in relation to different documents in terms of type and length, a total of 20 documents were evaluated using the pipeline. The selection is listed in Appendix Table 28.

The runtimes observed on this specific document set are presented in Figure 22. As illustrated, the runtime increases linearly with the number of pages, which fulfills the condition of an acceptable scaling behavior. Additionally, Figure 23a provides a visual representation of the average distribution of the runtime among the individual components of the pipeline. This figure demonstrates that the runtime is primarily influenced by the classification and quote extraction stages, as these are the points where LLMs are employed.

Table 23b additionally illustrates the average token consumption across the modules. Consistently, classification accounts for the majority of tokens consumed. It is important to note that the preprocessing was performed without the summarization step during these experiments. Therefore, no tokens were consumed within the preprocessing module.



(a) Avg. runtime distribution.

(b) Avg. token usage distribution.

Figure 23: (a) Observed average runtime (in seconds) distribution and (b) token distribution among the individual pipeline modules.

.

7 Discussion

As presented in Chapter 6, the findings of the evaluation allow for an initial assessment of the extent to which the individual components of the conceptual pipeline and their implementation meet the objective of automating the analysis of climate documents.

However, in order to evaluate the overall suitability of the proposed approach, it is necessary to consider the initially established requirements. These are discussed in the following section. Subsequently, the inherent limitations of the approach will be addressed.

7.1 Requirements

The requirements to the developed concept are delineated in Section 4.2, while the requirements to the specific use case for implementation are expressed in Section 7.1. The following section revisits both lists and presents a detailed examination of each of the individual items, with the objective of determining the extent to which the proposed solution is able to satisfy them. To summarize the findings, Table 19 lists the general requirements to the pipeline concept, while Table 20 outlines the fulfillment of requirements specific to the use case described in Chapter 5.1.

With regard to the requirements concerned with the pipeline architecture, the implemented solution is both capable of processing different input formats via the document parsing module implemented, which has been implemented with the Unstructured library (see Section 5.2.1). In particular, it is possible to assess PDFs, which are the prevalent publication format for policy documents. The implemented pipeline accepts local files via their path specification and documents provided via URL, thereby enabling direct online access, which is particularly advantageous for NDCs and LTS documents published by UNFCCC.

Regarding the pipeline output, the implemented spreadsheet solution has been designed with the specific intention of aligning with the requirements of the analysts responsible for the MNZ NDC Transport Tracker. In different settings, alternative output formats, such as JSON or YAML files, may prove to be a superior fit, facilitating integration into the intended publication formats, such as web-based dashboards.

However, it is always important to ensure that the output clearly indicates the source of the extracted quotes within the source document, thus facilitating its verification if required. In the implemented prototype, this is achieved by specifying the page number. The supplementary highlighted output document provides additional visual aid for orientation.

Several requirements are placed specifically on the extraction of quotes. The pipeline must be capable of extracting individual sentences containing relevant quotes from the document and verifying their verbatim presence in the document. Additionally, the pipeline should be able to extract multiple quotes from a single document, when appropriate. Both the conceptual and implemented pipeline fulfill this set of requirements through the quote extraction module, dedicated to this task.

A particular requirement is the processing of scanned documents, which occur with some regularity in the context of climate policy documents. Within the implemented pipeline, this task is addressed by the document parser, which integrates the OCR capacity of the Tesseract library via Unstructured. As a result, scanned text pages are treated in the pipeline analogously to other pages. In the conceptual pipeline, it is additionally possible to treat scanned objects as image elements. In this case, the information contained is extracted via prompting with multimodal LLMs during the summarization step.

Another significant challenge presented by climate policy documents consists in the frequent listing of relevant content in tabular format. As concluded in a study by Sui et al. (2024), general-purpose LLMs, such as the GPT models applied for implementation of the pipeline, have the basic capabilities to understand structural information in tabular format. The application, however, has shown that table-related tasks such as layout detection remains error-prone, specially for complex tables with heterogeneous substructures. Special importance comes to the selected serialization method, which in this case has been adapted to the task of verbatim quote extraction, as delineated in Section 5.2.3.

With regard to challenges concerning the language employed in the documents, it has been established that the pipeline should be capable of processing languages present in submissions to the UNFCCC. As detailed in Section 4.5, the capabilities of LLM diverge depending on the quantity of resources in the respective language contained in the training corpus. Robinson et al. (2023) differentiate between High Resource Languages (HRL) and Low Resource Languages (LRL) in this context. In view of the fact that, as pointed out in Section 4.5, the official UN languages pertain to the category of HRLs, the requirement can be considered as fulfilled.

The capabilities of the employed models for the understanding of climate policy domain-specific language has not been subject to studies. Nevertheless, as this capacity is a necessary condition for the functioning of the entire process, the positive outcomes of the evaluations in Chapter 6 demonstrate that the models employed are capable of adequately addressing the vocabulary present in the documents.

The pipeline was designed in a manner that allows for the implementation of adjustments to the applied indicators and taxonomy, which define the information of interest within the document. This enables the pipeline to be adapted in accordance with the characteristics of the individual use case. This approach was successfully implemented in the

Aspect	**Requirement**	**Pipeline Module**	**Fulfilled**
Architecture	Ability to process different document formats	Preprocessing	✓
	Acceptable scaling properties	Preprocessing, Retrieval, Quotation, Classification	✓
Architecture/ LLM	Ability to extract quotes from the document corpus	Quotation	✓
	Verbatim reproduction of the extracted text passages	Quotation	✓
	Ability to extract various quotes from a single document	Quotation	✓
	Multilingual LLM, at least covering the six official UN languages, or machine translation function	Preprocessing, Retrieval, Quotation, Classification	✓
	OCR capabilities included in the pipeline	Preprocessing	✓
LLM	LLM familiar with climate domain-specific and technical terminology	Preprocessing, Quotation, Classification	(✓)
	LLM is capable of processing tables	Preprocessing, Quotation	(✓)

Table 19: Fulfillment of the requirements set for the automated climate policy analysis pipeline.

prototype, as described in Chapter 5.2.4. Additionally, the use case-specific requirement for the assignment of multiple labels to a quote was established. The tagging methodology employed is capable of meeting this requirement (see Chapter 5.2.4).

Aspect	**Requirement**	**Pipeline Module**	**Fulfilled**
Architecture	Ability to process NDCs and LTSs published by UNFCCC	Preprocessing	✓
	Traceability of results in the original document	Preprocessing, Retrieval, Quotation, Classification	✓
	Alignment of output format with database structure	Classification	✓
Architecture/ LLM	Implementation of indicator set provided by MNZ	Classification	✓
	Multi-label classification possible	Classification	✓

Table 20: Fulfillment of the requirements set for the automated climate policy analysis pipeline within the specific use case.

7.2 Limitations

Despite the successful fulfillment of the established criteria, the proposed concept still exhibits a number of limitations. The following section presents a selection of the most significant limitations from the author's perspective.

Performance

The conducted evaluation of the individual steps in Chapter 6 reveals overall satisfactory results in terms of performance. However, certain weaknesses also became apparent. For instance, as illustrated in Table 13, the pipeline encounters significant challenges in classifying the conditionality of targets. Identifying conditions for targets is a particularly challenging task because the conditions are not always explicitly mentioned in direct connection with the target, nor situated in the immediate context of the target quotes. Instead, the conditions are often referenced centrally within the document, for example, at the beginning of a chapter. In addition, conditions may be deliberately concealed by document issuers. The extent to which the recognition of conditions can be conducted in a dedicated manner is a subject of further research in this domain.

Resource Efficiency

The clearest limitations of the implemented approach, when the entirety of the pipeline is taken into account, is the continued high consumption of resources. The runtime exhibits linear scaling behavior, even when processing particularly lengthy documents of up to 200 pages. However, the basic level of the runtime remains high, averaging approximately 4500 seconds (see Figure 23a), which is the equivalent of over an hour. A low runtime was not specified in the requirements, as the automated analysis can run in the background and therefore does not tie up the human analyst's time. Nevertheless, with regard to the usability of the pipeline, a shorter runtime would be advantageous and a goal for future development steps. The same applies for the potentially high costs of the pipeline due to the related token consumption (see Figure 23b).

Generalizability

The functionality of the pipeline has been evaluated exclusively within the context of the presented prototype implementation for the analysis of NDCs and LTSs. It remains to be seen whether the results are generalizable to a broader selection of climate policy documents. A question of particular interest is whether the pipeline is applicable to national documents. The large diversity of languages encountered within these documents presents a significant challenge to the approach of leveraging general-purpose LLMs without fine-tuning. In particular, documents in low-resource languages may limit the applicability of the proposed approach.

Dependency on Specific Technology

With regard to the implemented pipeline, it should be noted that the results are highly dependent on individual software packages, which could present a potential limitation. For instance, several tasks, including document parsing, OCR, chunking, and table serialization, are performed via the Unstructured library. This creates the risk of relying on packages which might become obsolete or incompatible in the future, potentially limiting the long-term viability and adaptability of the pipeline as new technologies emerge.

8 Summary

The following chapter provides a summary for the presented thesis. First, the most significant findings are summarized in a concluding statement (8.1). These findings serve as a foundation for the formulation of remaining open questions, providing insights into potential future research directions in the context of automated climate policy analysis using NLP (8.2).

8.1 Conclusion

The presented thesis describes the development of a concept for an automated climate policy analysis pipeline that employs NLPs to identify and classify statements in climate policy documents. The pipeline is designed to assist the labeling process carried out by human policy analysts, by automatic identification of quotes within the documents that represent claims on measures and targets to be undertaken by the respective country. Additionally, the pipeline is capable of classifying the targets through application of a predefined indicator set and a related taxonomy, by which the indicators are categorized.

To evaluate the performance of the concept, the pipeline was implemented as a prototype for the specific use case of identifying transport sector targets and measures in NDCs or LTSs for the NDC Transport Tracker by MNZ. The implemented pipeline introduced in Chapter 5 and the results of its evaluation laid out in Chapter 6 demonstrated the feasibility of executing intricate and domain-specific tasks, such as the automated detection and classification of quotes containing transport-related climate targets and measures, via the utilization of general-purpose LLMs, with satisfactory outcomes. To this end, the models were applied with selected prompting techniques from literature and their adoption was adjusted to the respective requirements of the document processing steps.

The entire process was conducted without the use of domain-adapted or fine-tuned models, thereby circumventing the necessity for additional work associated with data acquisition, annotation, and adaptation. The resulting framework is designed to be flexible and extensible from both a technical and a policy perspective. It has been developed in a platform-agnostic manner, allowing the employment of any existing open-source or commercial tool for each component in the pipeline. In particular, the approach is expandable to encompass as many languages as the applied models have been trained for. In light of the versatility of existing multi-lingual LLMs (Aryabumi et al., 2024), this paves the way for the automated analysis of a vast array of national policy documents, which may offer further potential for streamlining and enhancing the work of climate policy analysts.

8.2 Outlook

Although the implemented pipeline has already yielded valuable outcomes, some potential refinements and enhancements are conceivable within the various modules and components of the pipeline. These possibilities are subject to future research endeavors. The identified limitations outlined in Chapter 7.2 offer a basis for determining the direction of future work to this end.

With regard to improving the performance of the pipeline, two of the most significant outcomes of the evaluation in Chapter 6 were the identification of highly intricate indicators and the potential for enhancing the outcomes by incorporating examples from the annotated data set. In the future, it would be beneficial to investigate whether this approach also markedly enhances the classification of indicators that have already been effectively identified, and whether the enhancements justify the possible additional effort involved in obtaining the examples in scenarios without previous availability of an annotated data set.

In addition, the difficulty of identifying conditionality in target quotes was acknowledged. The incorporation of contextual information to the classification prompt did not yield any significant improvement in the limited sample size. It is essential to ascertain whether an expanded examination would yield more reliable outcomes. Alternatively, a dedicated search for conditional clauses within the document could be a viable approach, given that the conditions are frequently not situated in the immediate context of the target quotes.

Regarding resource utilization, the efficiency of the pipeline has the potential to be enhanced with regard to runtime, as a large share of the tasks are parallelizable. In the context of the work, only commercially available, proprietary LLMs (GPT 3.5 and GPT 4) were utilized. It remains to be seen whether publicly accessible, state-of-the-art open-source models, such as Llama 3 (Dubey et al., 2024) or Mixtral 8x7B (Jiang et al., 2023), are capable of fulfilling the involved set of tasks, thus offering the opportunity to save costs.

It is further conceivable that additional components may be incorporated into future iterations of the pipeline. For instance, the preprocessing module could be extended with the incorporation of a web crawler that directly accesses and scrapes the document files from a source platforms. This is specially worth considering, if the focal range is amplified to a larger diversity of policy documents. The crawler may be configured to run periodically, e.g., weekly, to ensure up-to-date coverage of published policy documents. The pipeline could also be expanded to include an additional component that accesses existing knowledge related to previously analyzed policy documents. In this way, countries' previous claims are included in the analysis of current policies, for example to check their consistency. Furthermore, this enables an automated comparison between documents across countries.

Finally, it should be noted that it is only through the acquisition of user experience that it will be possible to determine the extent to which the proposed pipeline facilitates the desired time savings and provides analytical support in the daily work of climate policy analysts. It is thus imperative that future research on this topic include studies on the practical application of the pipeline. The upcoming completion of the current 5-year cycle for the submission of updated NDCs in the context of COP 29 in Baku offers a suitable opportunity for this.

A Appendix

A.1 Indicators

Indicator	Description
T_Economy_Unc	An unconditional reduction target for greenhouse gas emissions (covering CO2 and other relevant greenhouse gases) has been set for the whole economy or collectively for all sectors covered in the document. This target is not conditional or contingent upon any other developments.
T_Economy_C	A conditional reduction target for greenhouse gas emissions (covering CO2 and other relevant greenhouse gases) has been set for the whole economy or collectively for all sectors covered in the document. This target is conditional or contingent upon other developments.
T_Transport_Unc	An unconditional greenhouse gas emission mitigation target has been set for transport or a transport subsector. This target is not conditional or contingent upon any other developments.
T_Transport_C	A conditional greenhouse gas emission mitigation target has been set for transport or a transport subsector. This target is conditional or contingent upon other developments.
T_Adaptation_Unc	An unconditional quantifiable target has been adopted for transport adaptation and resilience. This target is not conditional or contingent upon any other developments.
T_Adaptation_C	A conditional quantifiable target has been adopted for transport adaptation and resilience. This target is conditional or contingent upon other developments.
T_Transport_O_Unc	This category includes other unconditional non-emission targets focusing on transport, such as transport activity, mode share, air pollution, or energy consumption. This target is not conditional or contingent upon any other developments.

T_Transport_O_C	This category includes other conditional non-emission targets focusing on transport, such as transport activity, mode share, air pollution, or energy consumption. This target is conditional or contingent upon other developments.
T_Energy	A mitigation target for the energy sector (and thus indirectly for transport) has been adopted.
T_Netzero	This category captures any economy-wide net zero emission targets.

Table 21: Complete set of target indicators, as defined by Mobilize Net-Zero.

Indicator	**Description**
	Structural and Technical
R_System	This indicator identifies efforts to adapt to climate change impacts to transport infrastructure and to increase its resilience
R_Maintain	This indicator identifies general efforts to repair or maintain transport infrastructure, without reference to climate change adaptation.
R_Risk	This indicator identifies efforts to understand risks and impacts to the transport system (e.g. through modelling).
R_Tech	This indicator identifies efforts to adopt resilient transport technologies (e.g. climate resilient materials for streets or cars).
	Informational and Educational
R_Monitoring	This indicator identifies efforts to adopt monitoring systems, e.g. to detect risks early on.
R_Inform	This indicator identifies efforts to adopt notification systems, e.g. to inform drivers about flooding, so they can take alternate routes.
R_Emergency	This indicator records references to emergency and disaster planning that is specifically related to transport.

`R_Education`	This indicator records efforts to educate and train transport officials regarding the vulnerability of transport systems and infrastructure to climate change.
`R_Warning`	This indicator covers explicit mentions of an early warning system.
Institutional and Regulatory	
`R_Planning`	This subcategory records any activities designed to raise the importance of resilience and adaptation in transport planning.
`R_Relocation`	This subcategory refers to efforts to relocate infrastructure or populations due to current or anticipated threats.
`R_Redundancy`	This refers to the construction of redundant infrastructure/facilities, to prepare for the possible failure of existing systems.
`R_Disinvest`	This refers to any measures to discontinue or avoid expanding transport services or infrastructure.
`R_Laws`	This subcategory identifies laws or regulations that focus on climate change adaptation in the transport sector.
`R_Design`	This subcategory refers to the adoption of improved, more resilient design standards to effectively protect or reinforce transport facilities or infrastructure.
Other Adaptation Measures	
`R_Other`	Other adaptation measures for transport not falling under the categories listed above.

Table 22: Complete set of adaptation measure indicators, as defined by Mobilize Net-Zero.

Indicator	Description
Transport System Improvements	
`A_Complan`	This indicator records any general mention of transport planning.

`A_Natmobplan` A national mobility plan provides a comprehensive long-term planning framework for the transport sector. It features a vision and timeframes for action at the national level. This indicator records any activities that focus on nationwide transport.

`A_SUMP` A sustainable urban mobility plan (SUMP) is “a strategic plan designed to satisfy the mobility needs of people and businesses in cities and their surroundings for a better quality of life. It builds on existing planning practices and takes due consideration of integration, participation and evaluation principles”. If the document refers to a SUMP or an integrated approach on urban mobility, this is captured here.

`A_LATM` The measure looks at transport or a specific transport method with the goal of improving traffic flow.

`A_Landuse` This indicator captures land-use planning related to transport. The most important land use elements are density (of population and/or jobs), diversity (mix of uses), design (pedestrian quality, street network density), and distance to public transport.

`A_Density` This measure intends to develop compact cities by achieving high density, usage diversity and accessibility. It tries to avoid or reduce urban sprawl.

`A_Mixuse` “Mixed-use” is an urban planning approach that combines several urban functions (living, working and shopping) in close proximity. One goal is to encourage short travel distances.

`S_Infraimprove` This indicator is for measures that outline general improvements in transport infrastructure or the transport system as a whole, without providing details about specific measures.

S_Infraexpansion
Activities that aim to introduce new infrastructure or expand infrastructure for transport are captured by this indicator. If a measure is dedicated to a specific transport mode, then it might be captured under that specific indicator. Any general mention of expanding transport infrastructure is collected here. This can lead to rebound effects, in particular by causing an increase in traffic activity

S_Intermodality
Intermodality is the combination of different transport modes with the goal of enabling convenient, seamless transfer between them. Any general activities that highlight intermodality but do not specify specific actions are included here.

I_Freighteff
This indicator records general efficiency improvements in freight. If the document does not specify a specific activity or action that belongs to the other freight efficiency activities, then it is captured here.

I_Load
This indicator is for measures that encourage reliance on high-capacity vehicles (trains, ships, etc.) in order to achieve lower carbon intensity per ton transported.

S_Railfreight
In freight transport, intermodality often entails a shift to less carbon-intensive transport modes (e.g. rail and waterborne transport). This indicator records any actions that support a shift of road freight to rail or waterways.

I_Education
This indicator is for general educational activities and behavioral change related to transport, e.g. concerning the environmental impacts of private vehicle use, the benefits of electric vehicles, etc.

I_Ecodriving
Ecodriving refers to educational measures that encourage more efficient driving practices. Such practices can reduce fuel consumption.

I_Capacity — “Capacity building” refers in general to the training of individuals and organizations to acquire new skills and knowledge on a specific topic. “Sustainable transport capacity building” refers more specifically to the acquisition of skills to plan and implement safe, sustainable, affordable and resilient mobility systems.

I_Campaigns — This indicator captures mention of public awareness campaigns related to transport.

Mode Shift and Demand Management

A_TDM — This indicator records any general mention of activities focusing on reducing demand for motorized transport.

S_Parking — This measure refers to actions that aim to improve parking management.

A_Parkingprice — The demand for mobility can be influenced by introducing parking pricing. A fee for on-street parking can motivate people to leave their cars at home or to use park-and-ride facilities outside the city.

A_Caraccess — This indicator covers measures that restrict vehicle use in designated zones. One method is to restrict operation to certain weekdays based on license plate number.

A_Commute — This indicator refers to the management of circumstances and incentives for employee commuter travel and working arrangements to reduce traffic and automobile use.

A_Teleworking — This measure looks at implementing variable work hours in order to reduce traffic congestion during rush hour and reduce commuting times. Any mention of flexible work hours, alternative work approaches and staggered shifts are included under this indicator.

A_Economic — Economic instruments, taxes and subsidies are incentives to integrate environmental costs and benefits into the budgets of households and firms.

A_Emistrad An emissions trading system (ETS) or cap-and-trade system is a pricing mechanism for emitted greenhouse gas emissions. Unlike a direct carbon tax, where the unit price of CO2 is fixed, under an emissions trading scheme, the price per tonne of CO2 varies. The overall amount of emissions is fixed for a given period of time (e.g. annually). Entities are allocated a set amount of CO2 emissions allowances, or quotas, and trade emissions with each another. Those able to reduce their emissions below their allowance level can trade them with those emitting in excess of their allowance.

A_Finance This indicator records GHG emission reductions resulting from the use of any financial instruments. Financial instruments that aim to support decarbonization include climate finance solutions, investments in EVs, green bonds, etc.

A_Procurement Green procurement refers to activities by stakeholders to take environmental impacts into account when procuring goods and services. Applied to transport, it means that a public authority can develop green procurement regulations that, for example, only allow the purchase of zero-emission vehicles. Such measures can support the transition to cleaner public vehicle fleets and more sustainable consumption.

A_Fossilfuelsubs Energy subsidies are used by governments to lower the cost of producing or consuming fossil fuels. Eliminating such subsidies can help to reduce reliance on fossil fuels. This indicator records measures and actions in this area.

A_Fueltax National and state governments impose taxes on the sale of fuel. Every fuel type is taxed differently. One aim of higher fuel taxes is to reduce fuel consumption and encourage more efficient transport modes.

A_Vehicletax This indicator refers to measures that link taxes on vehicle purchase and ownership to carbon emissions.

A_Roadcharging This indicator refers to surcharges applied to general or specific road use, including in particular highway tolls. This includes congestion pricing.

S_PublicTransport This indicator covers all activities that aim to improve the public transport system.

S_PTIntegration Activities that aim to expand public transport or integrate different public transport services into a single system are covered under this indicator.

S_PTPriority This indicator looks at actions that give priority to public transport over other modes. Examples include transit signal priorities, access priority, intelligent transport systems and express lanes.

S_BRT Bus rapid transit (BRT) is a bus system with high speed, capacity, punctuality and operating flexibility. Common characteristics of a BRT system include the use of bus-only lanes, advance ticketing, and articulated buses.

S_Activemobility General measures that refer to walking and cycling are included here.

S_Walking Any action that specifically mentions improving walking.

S_Cycling Any action that specifically mentions improving cycling.

S_Sharedmob This indicator includes general measures in the area of shared mobility, such as bike sharing, car-sharing, shared scooters etc.

S_Ondemand This indicator identifies actions that support the implementation of collective on-demand services.

S_Maas Mobility-as-a-Service (MaaS) refers to solutions that integrate several mobility options through a digital platform that allows trips to be planned, booked and paid for. This is especially relevant for ticket integration and electronic payment.

I_Other This indicator includes activities that mention the use of innovation and digitalization to improve the efficiency of transport.

`I_ITS`	Intelligent transport systems harness technology to improve the management and operation of transport services. Relevant technologies include sensors, wireless communications, notification systems and other ICT solutions.
`I_Autonomous`	This indicator identifies measures that discuss self-driving vehicles, artificial intelligence and any other mechanisms that support the automation of passenger and freight transport.
`I_DataModelling`	This indicator identifies any measures related to transport data (e.g. collection, analysis or application) as well as models designed to predict traffic flows or transport demand growth.
Low-Carbon Fuels and Energy Vectors	
`I_Vehicleimprove`	This indicator identifies any general vehicle improvement measures that are included in the document.
`I_Fuelqualimprove`	A high-quality fuel contains very low levels of sulfur. Countries set fuel quality standards in order to guarantee fuel quality. This indicator covers any mention of clean fuels or better fuel quality in the transport sector.
`I_Inspection`	A well-maintained vehicle can ensure higher energy efficiency. This indicator considers measures that pertain to vehicle inspections or maintenance.
`I_Efficiencystd`	This indicator captures more stringent emission standards that regulate air pollution exhaust emission, such as the EURO standards Euro1-6.
`I_Vehicleeff`	This indicator captures measures designed to improve vehicle efficiency or lower transport emissions. This is done through fuel economy standards, which regulate how far a vehicle must travel when consuming a given quantity of fuel (e.g. in liters per 100 km or miles per gallon).

A_LEZ
Low emission zones are areas that limit vehicle operation based on their emission of pollutants. Such zones, which are often in urban areas, may prohibit such vehicles entirely or assess a fee.

I_VehicleRestrictions
This indicator encompasses various restrictions to vehicle ownership or purchase, including import bans on older vehicles or sale restrictions on particularly polluting vehicles.

I_Vehiclescrappage
In order to support the transition to cleaner, more efficient vehicles, governments may provide incentives when an owner scraps their current, old vehicle (rather than reselling it).

I_Lowemissionincentive
This indicator refers to purchase incentives granted to consumers for lower emission vehicles (excluding electric and hybrid vehicles).

I_Altfuels
Any general reference to the use of alternative fuels in the transport sector is recorded here.

I_Ethanol
Considered a renewable energy, ethanol can be blended with gasoline in order to reduce the carbon intensity of the fuel.

I_Biofuel
Conventional diesel and gasoline can be mixed with less carbon-intense fuels. Many national governments set blending mandates (for example, 10% or 20% of diesel has to be biofuel). Any general biofuel blending mandates are covered here.

I_LPGCNGLNG
This indicator is for measures that refer to liquified petroleum gas (LPG), compressed natural gas (CNG) or liquified natural gas (LNG) in the transport sector.

I_Hydrogen
A relatively new fuel in the transport sector, hydrogen is used in fuel-cell electric vehicles. Green hydrogen that is produced using renewable electricity is seen as one important component of the energy transition in transport.

I_RE	Renewable energy for transport looks at the use of biofuels, green hydrogen and green electricity. This indicator captures any actions that make a direct link between transport and renewables.
I_Transportlabel	This indicator refers to measures requiring publication of information on environment impacts. The goal is to create transparency about the greenhouse gas emissions caused by a vehicle, fuel or activity. This indicator encompasses any general measures that do not describe a specific labeling requirement.
I_Efficiencylabel	This indicator refers to labels that show the carbon intensity of new vehicles.
I_Freightlabel	This indicator identifies measures designed to introduce or expand the labeling of goods (for example, where they come from and how they have been transported).
I_Vehiclelabel	Vehicle labels are placed on new vehicles in order to show expected average fuel consumption.
I_Fuellabel	Fuel labels show the carbon intensity and quality of a given fuel.
Electrification	
I_Emobility	Any general policies that refer to electric mobility without specifying a transport mode or specific measure are covered by this indicator.
I_Emobilitycharging	Electric vehicle charging infrastructure is needed to promote the adoption of electric vehicles. Measures that seek to increase the number of public charging stations or facilitate more private/public charging points are covered here.
I_Smartcharging	Smart charging refers to systems that optimize electric vehicle charging by prioritizing off-peak hours or times of high variable renewable feed-in.

`I_Emobilitypurchase`	National and local governments can support the transition to e-mobility by providing financial incentives for the purchase of electric vehicles.
`I_ICEdiesel`	This indicator identifies efforts to phase-out of fossil fuel vehicles. The most common policy is a sales ban on new diesel or gasoline vehicles starting in a specific year. Such policies seek to accelerate the adoption of electric vehicles.
`S_Micromobility`	Micromobility refers to electric personal transportation devices, such as electric kick-scooters and other electric-powered devices, not covered under shared mobility.
	Innovation and Up-scaling
`I_Aviation`	Any general measures that focus on the aviation sector are referred to here.
`I_Aircraftfleet`	Newer aircraft are generally more energy efficient. This indicator refers to activities designed to renew the aircraft fleet or only allow newer aircraft to operate.
`I_CO2certificate`	CO2 certification systems aim to mitigate greenhouse gas emissions by airports and ground operations. This indicator is for initiatives designed to improve the energy efficiency and carbon footprint of airports.
`I_Capacityairport`	Constraints to airport development or operations are included here.
`I_Jetfuel`	This indicator refers to policies designed to lower the carbon intensity of fuels for aviation or to introduce alternative fuel sources, including biofuel blending mandates.
`I_Airtraffic`	Any measures that focus on improving air traffic are referred to here.
`I_Shipping`	This indicator refers to any general measures that target shipping, maritime transport or inland navigation.

`I_Onshorepower`	Support on-shore power and electric charging facilities in ports. While low-carbon fuels for ships are still being explored, there are already several solutions for providing electricity to the vessel when docked. This is also commonly known as “cold ironing”.
`I_PortInfra`	This indicator refers to improvements to ports and other shore-based facilities.
`I_Shipefficiency`	The indicator identifies actions that aim to improve the energy efficiency of ships.

Table 23: Complete set of mitigation measure indicators, as defined by Mobilize Net-Zero.

A.2 Prompts

Prompt

```
""" You are an assistant tasked with summarizing text from climate
policy documents.
Give a concise summary of the text chunk:  {element} """
```

Figure 24: Prompt employed for the summarization of text elements.

Prompt

```
""" You are an assistant tasked with summarizing tables from climate
policy documents.
Give a concise summary of the table:  {element} """
```

Figure 25: Prompt employed for the summarization of table elements.

Prompt

```
""" You are a useful bot that is especially good at OCR from images.
Given the image, provide the following information:
- A detailed description of the image.  Be specific about graphs,
such as bar plots.
- Classify which kind of information is contained in the image
(picture, graph, table, etc.)
- The text contained in the image.  If necessary, provide a concise
summary.
Image:  {encoded_image} """
```

Figure 26: Prompt employed for the summarization of images.

Prompt

```
""" You are a climate and transport policy analyst, specialized in
Nationally Determined Contributions (NDCs).  You will be provided
with climate policy document snippets from NDCs.  Your task is to
extract quotes from these sippets.
The quotes should define targets, measures and or actions undertaken
or planned.
If the snippet does not contain relevant information, return an
empty answer.  Formulate your response ONLY based on the information
contained in the document snippet.
BEFORE responding, CHECK whether the quote reproduces the EXACT
wording from the document.
Answer providing a JSON structure, where each key-value pair
represents one quote.
Here is the NDC snippet:  {Context}
{format_instructions} """
```

Figure 27: Prompt employed for the extraction of quotes from elements.

Prompt

```
""" You are a climate and transport policy analyst, specialized in
Nationally Determined Contributions (NDCs).
You will be provided with a list of quotes from a climate policy
document, like an NDCs.
Your task is to select those quotes, that define targets, measures
and or actions undertaken or planned.
The selected targets, measures or actions must be either from the
transport or the energy sector, or define economy-wide greenhouse
gas (GHG) reduction targets.
Do NOT alter any of the quotes.
In case the input list is empty, respond with an empty answer as
well.
Here is the NDC snippet:  {Quotes}
{format_instructions} """
```

Figure 28: Prompt employed for the revision of the extracted quotes.

Prompt

```
""" You are an expert professor specialized in grading students'
answers to questions.
The students' answer was based on the following context information:
{snippet}
He was asked to extract quotes from this context. Here is the real
answer:
{reference}
The student may answer with a list of possible quotes. You are
grading 'CORRECT' if the real answer is contained in the predicted
answers, or if the student does not provide any answer and the
reference is an empty string:
{prediction}
Respond with CORRECT or INCORRECT """
```

Figure 29: Prompt employed for the LLM-judge to evaluate the quote extraction performance.

Prompt

```
""" Extract the desired information from the following quote.
Only extract the properties mentioned in the 'ResultObject'
function.
Quote: {quote} """
```

Figure 30: Prompt employed for the classification of quotes.

Prompt

```
""" You are a climate and transport policy analyst, specialized
in Nationally Determined Contributions (NDCs). Your task is to
classify text from such climate policy documents. You will be
provided with quotes from NDCs.
Use the following step-by-step instructions to classify the quotes:
Step 1 - Does the quote define a target? If yes, classify as ‘‘T’’,
if not, STOP the process and answer with an explanation, why the
quote does not describe any target.
Step 2 - Which sector is the target concerning about? If it
concerns the Energy sector, classify as ‘‘T_Energy’’ and STOP
the process. If it concerns the Transport sector, classify as
‘‘T_Transport’’, if it concerns an economy-wide target, classify as
‘‘T_Economy’’. If it concerns any other sector, STOP the process and
answer naming the sector.
Step 3 - If the classified sector is Transport, detect if the
target is greenhouse gas (GHG) related or not. If it is not GHG
related, classify as ‘‘T_Transport_O’’. If it is GHG related, keep
the classification as ‘‘T_Transport’’.
Step 4 - If the classified sector is Transport, detect if the
target is climate change mitigation related or climate change
adaptation related. If it is related to adaptation, classify
as ‘‘T_Adaptation’’. If it is related to mitigation, keep the
classification as ‘‘T_Transport’’.
Step 5 - Does the quote contain any conditionality related to the
defined target? If yes, end the classification with a suffix ‘‘_C’’.
If there is no conditionality defined for the target, add the suffix
‘‘_Unc’’.
Answer by ONLY returning the suggested complete classification.
Possible Answers: T_Adaptation_C, T_Adaptation_Unc, T_Economy_C,
T_Economy_Unc, T_Energy, T_Transport_C, T_Transport_O_C,
T_Transport_O_Unc, or T_Transport_Unc.
Quote: {quote} """
```

Figure 31: Single prompt employed for the classification of targets.

A.3 Retrieval

Initial Keywords
❒ "Transport"
❒ "Energy"
❒ "Net Zero"
❒ "Mitigation"
❒ "Adaptation"
❒ "Conditional"
❒ "Target"
❒ "Measure"

Table 24: Keywords initially employed for retrieval.

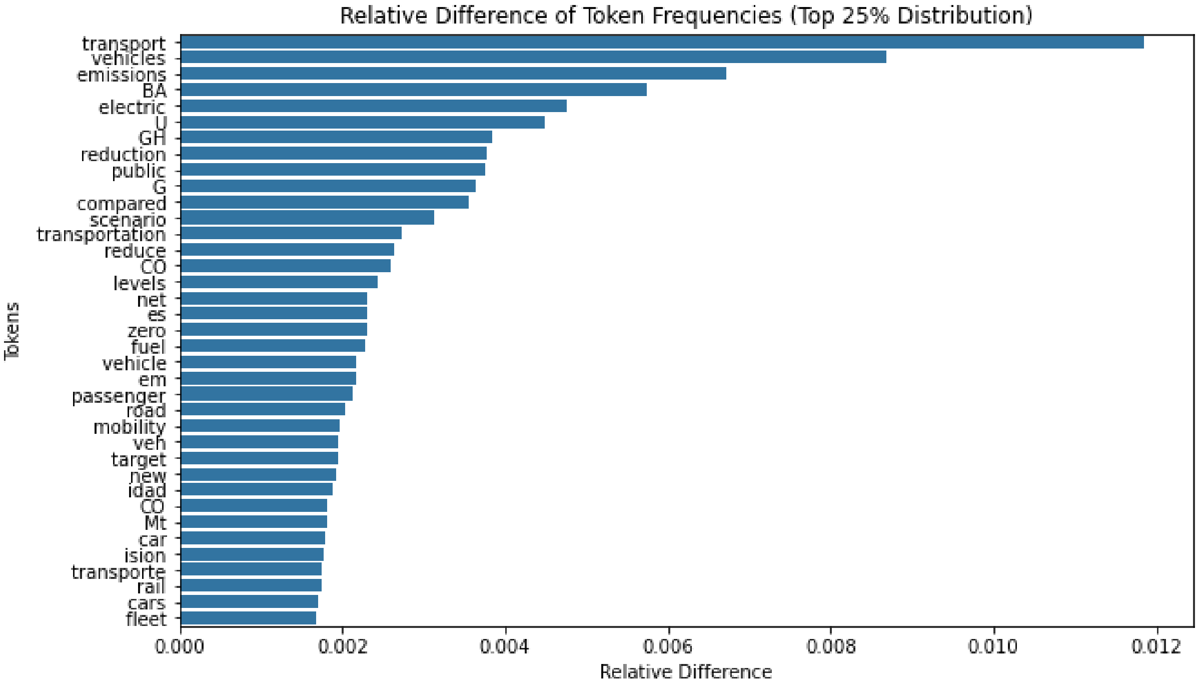


Figure 32: Distribution of frequent tokens within quotes containing target and measure information, corrected for general frequent tokens in climate policy documents. Excerpt limited to the top 25% of distribution mass.

A.4 Classification

Field	Description	Values
energy	Does the text refer to the ENERGY sector?	`True`, `False`
`transport`	Does the text refer to the TRANSPORT sector or a TRANSPORT-RELATED subsector?	`True`, `False`
`economy_wide`	Does the text not refer to a specific sector, but to the ECONOMY in general?	`True`, `False`
`mitigation`	Does the text concern climate change MITIGATION in a direct or indirect manner?	`True`, `False`
`adaptation`	Does the text concern climate change ADAPTATION or building resilience against consequences of climate change?	`True`, `False`
`ghg`	Does the text concern GREENHOUSE GAS EMISSIONS in a direct manner?	`True`, `False`
`net_zero`	Does the text define a NET-ZERO ghg emission target?	`True`, `False`
`conditional`	Does the text define a CONDITIONALITY or contingency upon other developments?	`True`, `False`
`unconditional`	This text does NOT define a CONDITIONALITY or contingency upon any other developments.	`True`, `False`

Table 25: Visualization of `TargetObject` Pydantic Model

A.5 Evaluation

Indicator	Precision (%)	Recall (%)	F_1-Score (%)
R_Disinvest	0.00	0.00	0.00
R_Relocation	100.00	33.30	50.00
R_Redundancy	0.00	0.00	0.00
R_Maintain	100.00	100.00	100.00
R_Laws	100.00	66.70	80.00
R_Monitoring	100.00	100.00	100.00
R_Emergency	100.00	100.00	100.00
R_Education	100.00	33.30	50.00
R_Inform	100.00	33.30	50.00
R_Tech	100.00	75.00	85.70
R_Other	0.00	0.00	0.00
R_Design	100.00	100.00	100.00
R_Risk	100.00	100.00	100.00
R_System	100.00	50.00	66.70
R_Warning	100.00	75.00	85.70
R_Planning	100.00	80.00	88.90

Table 26: Adaptation measure classification performance on individual indicators.

Indicator	**Precision (%)**	**Recall (%)**	**F_1-Score (%)**
I_Emobilitycharging	0.00	0.00	0.00
I_Freighteff	0.00	0.00	0.00
I_Inspection	0.00	0.00	0.00
I_Onshorepower	0.00	0.00	0.00
I_PortInfra	0.00	0.00	0.00
I_Shipefficiency	0.00	0.00	0.00
I_Campaigns	100.00	100.00	100.00
I_Shipping	0.00	0.00	0.00
I_Vehiclescrappage	0.00	0.00	0.00
S_Infraexpansion	0.00	0.00	0.00
S_Infraimprove	0.00	0.00	0.00
S_Ondemand	100.00	100.00	100.00
S_PTPriority	0.00	0.00	0.00
I_Transportlabel	0.00	0.00	0.00
I_CO2certificate	0.00	0.00	0.00
I_Fuelqualimprove	0.00	0.00	0.00
I_Altfuels	0.00	0.00	0.00
I_Aviation	0.00	0.00	0.00
A_LEZ	0.00	0.00	0.00
A_Finance	0.00	0.00	0.00
A_Density	0.00	0.00	0.00
A_Commute	0.00	0.00	0.00
A_Procurement	0.00	0.00	0.00
A_Caraccess	100.00	100.00	100.00
A_Vehicletax	0.00	0.00	0.00

A_Teleworking	0.00	0.00	0.00
I_Other	0.00	0.00	0.00
A_Natmobplan	100.00	50.00	66.67
I_Ecodriving	0.00	0.00	0.00
I_Vehiclerestrictions	100.00	50.00	66.67
I_Vehicleimprove	100.00	50.00	66.67
S_PTIntegration	0.00	0.00	0.00
S_Activemobility	0.00	0.00	0.00
S_Intermodality	100.00	33.33	50.00
S_Cycling	0.00	0.00	0.00
S_Railfreight	100.00	33.33	50.00
I_Vehicleeff	100.00	33.33	50.00
I_Capacity	100.00	66.67	80.00
A_LATM	0.00	0.00	0.00
S_Walking	0.00	0.00	0.00
A_Mixuse	0.00	0.00	0.00
A_SUMP	100.00	25.00	40.00
A_Complan	100.00	25.00	40.00
A_TDM	100.00	25.00	40.00
S_Parking	0.00	0.00	0.00
A_Landuse	100.00	20.00	33.33
S_Sharedmob	100.00	20.00	33.33
S_PublicTransport	100.00	16.67	28.57
I_Emobility	100.00	12.50	22.22

Table 27: Mitigation measure classification performance on individual indicators.

Country	**Type**	**Page Number**
Afghanistan	NDC	8
Armenia	LTS	37
Australia	NDC	7
Bhutan	LTS	189
Bosnia and Herzegovina	LTS	159
Brunei Darussalam	NDC	13
Cook Islands	NDC	3
Ethiopia	LTS	108
Equatorial Guinea	LTS	40
Indonesia	LTS	156
Ireland	LTS	90
Kenya	NDC	20
Namibia	NDC	44
New Zealand	LTS	68
Paraguay	NDC	6
Peru	NDC	12
South Sudan	NDC	9
Sri Lanka	LTS	124
Turkmenistan	NDC	60
UAE	NDC	5
Vanuatu	LTS	78
Yemen	NDC	16

Table 28: List of documents employed for pipeline evaluation.

References

Abudu, H., Wesseh, P. K., & Lin, B. (2024). Climate bonds toward achieving net zero emissions and carbon neutrality: Evidence from machine learning technique. *Journal of Management Science and Engineering*, *9*(1), 1–15. https://doi.org/10.1016/j.jmse.2023.10.001

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., ... Zoph, B. (2024, March). GPT-4 Technical Report. https://doi.org/10.48550/arXiv.2303.08774

Aitken, J. A., Rao, D. W., Alaybek, B., Sprenger, A., Mika, G., ... Leets, L. (2022). AI-Based Text Analysis for Evaluating Food Waste Policies. *AAAI 2022 Fall Symposium: The Role of AI in Responding to Climate Challenges*. https://www.climatechange.ai/papers/aaaifss2022/1

Altman, N., & Krzywinski, M. (2018). The curse(s) of dimensionality. *Nature Methods*, *15*(6), 399–400. https://doi.org/10.1038/s41592-018-0019-x

Anil, R., Borgeaud, S., Alayrac, J.-B., Yu, J., Soricut, R., ... Vinyals, O. (2024, June). Gemini: A Family of Highly Capable Multimodal Models. https://doi.org/10.48550/arXiv.2312.11805

Anthropic. (2024). The Claude 3 Model Family: Opus, Sonnet, Haiku. https://api.semanticscholar.org/CorpusID:268232499

Arora, S., Li, Y., Liang, Y., Ma, T., & Risteski, A. (2018, December). Linear Algebraic Structure of Word Senses, with Applications to Polysemy. Retrieved May 16, 2024, from http://arxiv.org/abs/1601.03764

Arowosegbe, A., & Oyelade, T. (2023). Application of Natural Language Processing (NLP) in Detecting and Preventing Suicide Ideation: A Systematic Review. *International Journal of Environmental Research and Public Health 2023, Vol. 20, Page 1514*, *20*(2), 1514. https://doi.org/10.3390/IJERPH20021514

Aryabumi, V., Dang, J., Talupuru, D., Dash, S., Cairuz, D., ... Hooker, S. (2024, May). Aya 23: Open Weight Releases to Further Multilingual Progress. https://doi.org/10.48550/arXiv.2405.15032

Batini, C., & Scannapieco, M. (2006). *Data Quality: Concepts, Methodologies and Techniques*. Springer Berlin Heidelberg.

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, *5*(2), 157–166. https://doi.org/10.1109/72.279181

Bengio, Y., Ducharme, R., Vincent, P., & Janvin, C. (2003). A Neural Probabilistic Language Model. *J. Mach. Learn. Res.* Retrieved June 28, 2024, from https://www.semanticscholar.org/paper/A-Neural-Probabilistic-Language-Model-Bengio-Ducharme/6c2b28f9354f667cd5bd07afc0471d8334430da7

Biesbroek, R., Badloe, S., & Athanasiadis, I. N. (2020). Machine learning for research on climate change adaptation policy integration: An exploratory UK case study. *Regional Environmental Change*, *20*(3), 85. https://doi.org/10.1007/s10113-020-01677-8

Bodansky, D. (2016). *THE PARIS CLIMATE CHANGE AGREEMENT: A NEW HOPE?* (Tech. rep.). http://climateactiontracker.org/global.html

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., … Amodei, D. (2020). Language Models are Few-Shot Learners. Retrieved January 28, 2024, from https://commoncrawl.org/the-data/

Cação, F. N., Costa, A. H. R., Unterstell, N., Yonaha, L., Stec, T., & Ishisaki, F. (2021). DeepPolicyTracker: Tracking Changes In Environmental Policy In The Brazilian Federal Official Gazette With Deep Learning. *ICML 2021 Workshop on Tackling Climate Change with Machine Learning*. https://www.climatechange.ai/papers/icml2021/35

Callaghan, M., Schleussner, C. F., Nath, S., Lejeune, Q., Knutson, T. R., … Minx, J. C. (2021). Machine-learning-based evidence and attribution mapping of 100,000 climate impact studies. *Nature Climate Change 2021 11:11*, *11*(11), 966–972. https://doi.org/10.1038/s41558-021-01168-6

Calvin, K., Dasgupta, D., Krinner, G., Mukherji, A., Thorne, P. W., … Péan, C. (2023, July). *IPCC, 2023: Climate Change 2023: Synthesis Report. Contribution of Working Groups I, II and III to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change [Core Writing Team, H. Lee and J. Romero (eds.)]. IPCC, Geneva, Switzerland.* (tech. rep.). Intergovernmental Panel on Climate Change. https://doi.org/10.59327/IPCC/AR6-9789291691647

Carroll, P., Singh, B., & Mangina, E. (2024). Uncovering gender dimensions in energy policy using Natural Language Processing. *Renewable and Sustainable Energy Reviews*, *193*, 114281. https://doi.org/10.1016/j.rser.2024.114281

Casas, M. J. d. V., Höhne, N., Mooldijk, S., Smit, S., Weishaupt, M., … Fallasch, F. (2021, July). *NDC Design - Systematic Analysis*. Umweltbundesamt. Retrieved July 31, 2024, from https://www.umweltbundesamt.de/en/publikationen/NDC-Design

Chakraborty, A. (2023, November). Challenges of LLM for Large Document Summarization : Exploring different LangChain approaches. Retrieved July 23, 2024, from https://medium.com/google-cloud/langchain-chain-types-large-document-summarization-using-langchain-and-google-cloud-vertex-ai-1650801899f6

Chen, H., Wu, L., Chen, J., Lu, W., & Ding, J. (2022). A comparative study of automated legal text classification using random forests and deep learning. *Information Processing & Management*, *59*(2), 102798. https://doi.org/10.1016/j.ipm.2021.102798

Chen, W. (2023, May). Large Language Models are few(1)-shot Table Reasoners. In A. Vlachos & I. Augenstein (Eds.), *Findings of the Association for Computational*

Linguistics: EACL 2023 (pp. 1120–1130). Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.findings-eacl.83

Clark, J., Wan, M., & Santos Rodriguez, R. (2023). Understanding Climate Legislation Decisions with Machine Learning. *NeurIPS 2023 Workshop on Tackling Climate Change with Machine Learning.* https://www.climatechange.ai/papers/neurips2023/121

Corringham, T., Spokoyny, D., Xiao, E., Cha, C., Lemarchand, C., ... Gershunov, A. (2021). BERT Classification of Paris Agreement Climate Action Plans. *Climate Change AI.* Retrieved March 31, 2024, from https://www.climatechange.ai/papers/icml2021/45

Dai, A. M., & Le, Q. V. (2015, November). Semi-supervised Sequence Learning. https://doi.org/10.48550/arXiv.1511.01432

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019, May). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. https://doi.org/10.48550/arXiv.1810.04805

Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., ... Zhao, Z. (2024, July). The Llama 3 Herd of Models. https://doi.org/10.48550/arXiv.2407.21783

European Parliament. (2020, September). What is artificial intelligence and how is it used? Retrieved June 3, 2024, from https://www.europarl.europa.eu/topics/en/article/20200827STO85804/what-is-artificial-intelligence-and-how-is-it-used

Fang, X., Xu, W., Tan, F. A., Zhang, J., Hu, Z., ... Faloutsos, C. (2024, February). Large Language Models(LLMs) on Tabular Data: Prediction, Generation, and Understanding – A Survey. https://doi.org/10.48550/arXiv.2402.17944

Firebanks-Quevedo, D., Planas, J., Buckingham, K., Taylor, C., Silva, D., ... Zamora-Cristales, R. (2022). Using machine learning to identify incentives in forestry policy: Towards a new paradigm in policy analysis. *Forest Policy and Economics*, *134*, 102624. https://doi.org/10.1016/j.forpol.2021.102624

Firth, J. R. (1957). *Studies in Linguistic Analysis.* Oxford: Blackwell. Retrieved May 15, 2024, from https://languagelog.ldc.upenn.edu/myl/Firth1957.pdf

Fleuret, F. (2023). The Little Book of Deep Learning.

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., ... Wang, H. (2024, March). Retrieval-Augmented Generation for Large Language Models: A Survey. https://doi.org/10.48550/arXiv.2312.10997

Ge, L., & Moh, T.-S. (2017). Improving text classification with word embedding. *2017 IEEE International Conference on Big Data (Big Data)*, 1796–1805. https://doi.org/10.1109/BigData.2017.8258123

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning.* The MIT Press. https://www.deeplearningbook.org/

Grootendorst, M. (2022, March). BERTopic: Neural topic modeling with a class-based TF-IDF procedure. https://doi.org/10.48550/arXiv.2203.05794

Grundmann, R. (2022). Using large text news archives for the analysis of climate change discourse: Some methodological observations. *Journal of Risk Research*, *25*(3), 395–406. https://doi.org/10.1080/13669877.2021.1894471

Gururangan, S., Marasović, A., Swayamdipta, S., Lo, K., Beltagy, I., … Smith, N. A. (2020, May). Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. https://doi.org/10.48550/arXiv.2004.10964

Han, J., Kamber, M., & Pei, J. (2012). *Data Mining - Concepts And Techniques 3rd Edition*. Elsevier. Retrieved May 6, 2024, from http://archive.org/details/the-morgan-kaufmann-series-in-data-management-systems-jiawei-han-micheline-kambe

Hätönen, V., & Melzer, F. (2021). From Talk to Action with Accountability: Monitoring the Public Discussion of Policy Makers with Deep Neural Networks and Topic Modelling. *Climate Change AI*. Retrieved July 17, 2024, from https://www.climatechange.ai/papers/icml2021/75

Hegselmann, S., Buendia, A., Lang, H., Agrawal, M., Jiang, X., & Sontag, D. (2023). TabLLM: Few-shot Classification of Tabular Data with Large Language Models. *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, 5549–5581. Retrieved July 24, 2024, from https://proceedings.mlr.press/v206/hegselmann23a.html

Herzig, J., Nowak, P. K., Müller, T., Piccinno, F., & Eisenschlos, J. M. (2020). TAPAS: Weakly Supervised Table Parsing via Pre-training. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 4320–4333. https://doi.org/10.18653/v1/2020.acl-main.398

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Hofmann, V., Pierrehumbert, J., & Schütze, H. (2021, August). Dynamic Contextualized Word Embeddings. In C. Zong, F. Xia, W. Li, & R. Navigli (Eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (pp. 6970–6984). Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.acl-long.542

Holtzman, A., Buys, J., Du, L., Forbes, M., & Choi, Y. (2020, February). The Curious Case of Neural Text Degeneration. https://doi.org/10.48550/arXiv.1904.09751

Hong, Z., Ward, L., Chard, K., Blaiszik, B., & Foster, I. (2021). Challenges and Advances in Information Extraction from Scientific Literature: A Review. *JOM*, *73*, 1–18. https://doi.org/10.1007/s11837-021-04902-9

Hu, Y., & Lu, Y. (2024, April). RAG and RAU: A Survey on Retrieval-Augmented Language Model in Natural Language Processing. https://doi.org/10.48550/arXiv.2404.19543

Huang, Y., Lv, T., Cui, L., Lu, Y., & Wei, F. (2022, July). LayoutLMv3: Pre-training for Document AI with Unified Text and Image Masking. https://doi.org/10.48550/arXiv.2204.08387

International Transport Forum. (2018, October). *Transport CO2 and the Paris Climate Agreement: Reviewing the Impact of Nationally Determined Contributions* (tech. rep.). OECD. Paris. https://doi.org/10.1787/23513b77-en

IPCC. (2022, June). *Global Warming of 1.5°C: IPCC Special Report on Impacts of Global Warming of 1.5°C above Pre-industrial Levels in Context of Strengthening Response to Climate Change, Sustainable Development, and Efforts to Eradicate Poverty* (1st ed.). Cambridge University Press. https://doi.org/10.1017/9781009157940

ISO/IEC 25012:2008. (2008). *Software Engineering – Software Product Quality Requirements and Evaluation (SQuaRE) – data quality model* (Standard). International Organization for Standardization. Geneva, CH.

Jalalzadeh Fard, B., Hasan, S. A., & Bell, J. E. (2022). CliMedBERT: A Pre-trained Language Model for Climate and Health-related Text. *Climate Change AI*. Retrieved March 31, 2024, from https://www.climatechange.ai/papers/neurips2022/110

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Singh Chaplot, D., … El Sayed, W. (2023). Mistral 7B.

Jin, Y., Li, J., Liu, Y., Gu, T., Wu, K., … Ma, L. (2024, May). Efficient Multimodal Large Language Models: A Survey. https://doi.org/10.48550/arXiv.2405.10739

Josefsson, S. (2006, October). *The Base16, Base32, and Base64 Data Encodings* (Request for Comments No. RFC 4648). Internet Engineering Task Force. https://doi.org/10.17487/RFC4648

Joshi, M., Chen, D., Liu, Y., Weld, D. S., Zettlemoyer, L., & Levy, O. (2020, January). SpanBERT: Improving Pre-training by Representing and Predicting Spans. https://doi.org/10.48550/arXiv.1907.10529

Juhasz, M., Marchand, T., Melwani, R., Dutia, K., Goodenough, S., … Franks, H. (2024, April). Identifying Climate Targets in National Laws and Policies using Machine Learning. https://doi.org/10.48550/arXiv.2404.02822

Jurafsky, D., & Martin, J. H. (2023). *Speech and Language Processing An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition* (Third Edition draft).

Kasami, T. (1965). An Efficient Recognition and Syntax-Analysis Algorithm for Context-Free Languages.

Khurana, D., Koli, A., Khatter, K., & Singh, S. (2023). Natural language processing: State of the art, current trends and challenges. *Multimedia Tools and Applications*, *82*(3), 3713–3744. https://doi.org/10.1007/S11042-022-13428-4

Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., & Iwasawa, Y. (2023, January). Large Language Models are Zero-Shot Reasoners. https://doi.org/10.48550/arXiv.2205.11916

Kölbel, J. F., Leippold, M., Rillaerts, J., & Wang, Q. (2022, July). Ask BERT: How Regulatory Disclosure of Transition and Physical Climate Risks affects the CDS Term Structure. https://doi.org/10.2139/ssrn.3616324

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84–90. https://doi.org/10.1145/3065386

Lareyre, F., Nasr, B., Chaudhuri, A., Di Lorenzo, G., Carlier, M., & Raffort, J. (2023). Comprehensive Review of Natural Language Processing (NLP) in Vascular Surgery. *EJVES Vascular Forum*, *60*, 57–63. https://doi.org/10.1016/J.EJVSVF.2023.09.002

Lee, R. S. T. (2024). *Natural Language Processing: A Textbook with Python Implementation*. Springer Nature Singapore. https://doi.org/10.1007/978-981-99-1999-4

Leggett, J. A. (2020). *The United Nations Framework Convention on Climate Change, the Kyoto Protocol, and the Paris Agreement: A Summary* (tech. rep. No. Technical Report R46204). Congressional Research Service. https://crsreports.congress.gov/

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., ... Zettlemoyer, L. (2019, October). BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. Retrieved July 1, 2024, from http://arxiv.org/abs/1910.13461

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., ... Kiela, D. (2021, April). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. Retrieved July 6, 2024, from http://arxiv.org/abs/2005.11401

Li, J., Chen, X., Hovy, E., & Jurafsky, D. (2016). Visualizing and Understanding Neural Models in NLP. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 681–691. https://doi.org/10.18653/v1/N16-1082

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., ... Stoyanov, V. (2019, July). RoBERTa: A Robustly Optimized BERT Pretraining Approach. https://doi.org/10.48550/arXiv.1907.11692

Luccioni, A., Baylor, E., & Duchene, N. (2020). Analyzing Sustainability Reports Using Natural Language Processing. http://arxiv.org/abs/2011.08073

Madaan, A., Tandon, N., Gupta, P., Hallinan, S., Gao, L., … Clark, P. (2023, May). Self-Refine: Iterative Refinement with Self-Feedback. https://doi.org/10.48550/arXiv.2303.17651

Mandl, C. E. (2019). Language, Syntax, and Semantics for Describing Dynamics of Systems. In C. E. Mandl (Ed.), *Managing Complexity in Social Systems: Leverage Points for Policy and Strategy* (pp. 41–58). Springer International Publishing. https://doi.org/10.1007/978-3-030-01645-6_5

Manning, C. D., & Schütze, H. (1999). *Foundations of statistical natural language processing*. MIT Press.

Maslej, N., Etchemendy, J., Ligett, K., Fattorini, L., Perrault, R., … Clark, J. (2024, April). *The AI Index 2024 Annual Report* (tech. rep.). AI Index Steering Committee, Institute for Human-Centered AI, Stanford University. Stanford, CA.

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, *5*. https://doi.org/https://doi.org/10.1007/BF02478259

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013, September). Efficient Estimation of Word Representations in Vector Space. Retrieved May 16, 2024, from http://arxiv.org/abs/1301.3781

Mikolov, T., Karafiát, M., Burget, L., Černocký, J., & Khudanpur, S. (2010). Recurrent neural network based language model. *Interspeech 2010*, 1045–1048. https://doi.org/10.21437/Interspeech.2010-343

Mitchell, T. M. (2013). *Machine learning* (Nachdr.). McGraw-Hill.

Murphy, K. P. (2022). *Probabilistic machine learning: An introduction*. The MIT Press.

Nussbaum, Z., Morris, J. X., Duderstadt, B., & Mulyar, A. (2024, February). Nomic Embed: Training a Reproducible Long Context Text Embedder. Retrieved May 16, 2024, from http://arxiv.org/abs/2402.01613

Olsson, C., Elhage, N., Nanda, N., Joseph, N., DasSarma, N., … Olah, C. (2022). In-context Learning and Induction Heads. *Transformer Circuits Thread*.

OpenAI. (2022). New and improved embedding model. Retrieved July 30, 2024, from https://openai.com/index/new-and-improved-embedding-model/

Pauw, W. P., Beck, T., & Valverde, M. J. (2022, February). NDC Explorer. https://doi.org/10.23661/ndc_explorer_2022_4.0

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543. https://doi.org/10.3115/v1/D14-1162

Petangoda, J., Deisenroth, M. P., & Monk, N. A. M. (2021, July). Learning to Transfer: A Foliated Theory. https://doi.org/10.48550/arXiv.2107.10763

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., ... Zettlemoyer, L. (2018, March). Deep contextualized word representations. https://doi.org/10.48550/arXiv.1802.05365

Planas, J., Catalonia Daniel Firebanks-Quevedo, O., Naydenova, G., Japan Ramansh Sharma, O., Taylor, C., ... Sharma, R. (2021). Beyond modeling: NLP Pipeline for efficient environmental policy analysis; Beyond modeling: NLP Pipeline for efficient environmental policy analysis. https://doi.org/10.1145/nnnnnnn.nnnnnnn

Pukelis, L., Bautista-Puig, N., Statulevičiūtė, G., Stančiauskas, V., Dikmener, G., & Akylbekova, D. (2022, November). OSDG 2.0: A multilingual tool for classifying text data by UN Sustainable Development Goals (SDGs). https://doi.org/10.48550/arXiv.2211.11252

Pukelis, L., Puig, N. B., Skrynik, M., & Stanciauskas, V. (2020, May). OSDG – Open-Source Approach to Classify Text Data by UN Sustainable Development Goals (SDGs). https://doi.org/10.48550/arXiv.2005.14569

Radford, A., & Narasimhan, K. (2018). Improving Language Understanding by Generative Pre-Training. Retrieved June 30, 2024, from https://www.semanticscholar.org/paper/Improving-Language-Understanding-by-Generative-Radford-Narasimhan/cd18800a0fe0b668a1cc19f2ec95b5003d0a5035

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., ... Liu, P. J. (2019). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. https://doi.org/10.48550/arXiv.1910.10683

Rao, D., & McMahan, B. (2019). *Natural language processing with PyTorch: Build intelligent language applications using deep learning* (First edition). O'Reilly Media.

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing.* https://arxiv.org/abs/1908.10084

Riedel, S., Douwe Kiela, Patrick Lewis, & Aleksandra Piktus. (2020, September). Retrieval Augmented Generation: Streamlining the creation of intelligent natural language processing models. Retrieved July 6, 2024, from https://ai.meta.com/blog/retrieval-augmented-generation-streamlining-the-creation-of-intelligent-natural-language-processing-models/

Robinson, N., Ogayo, P., Mortensen, D. R., & Neubig, G. (2023, December). ChatGPT MT: Competitive for High- (but Not Low-) Resource Languages. In P. Koehn, B. Haddow, T. Kocmi, & C. Monz (Eds.), *Proceedings of the Eighth Conference on Machine Translation* (pp. 392–418). Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.wmt-1.40

Rolnick, D., Donti, P. L., Kaack, L. H., Kochanski, K., Lacoste, A., ... Bengio, Y. (2023). Tackling Climate Change with Machine Learning. *ACM Computing Surveys*, *55*(2). https://doi.org/10.1145/3485128

Rudolph, M., & Blei, D. (2018). Dynamic Embeddings for Language Evolution. *Proceedings of the 2018 World Wide Web Conference on World Wide Web - WWW '18*, 1003–1011. https://doi.org/10.1145/3178876.3185999

Russell, S. J., Norvig, P., Chang, M.-w., Devlin, J., Dragan, A., ... Wooldridge, M. J. (2022). *Artificial intelligence: A modern approach* (Fourth edition, global edition). Pearson.

Safaei, M., & Longo, J. (2023). The End of the Policy Analyst? Testing the Capability of Artificial Intelligence to Generate Plausible, Persuasive, and Useful Policy Analysis. *Digital Government: Research and Practice*. https://doi.org/10.1145/3604570

Salamone, L. (2021, April). What is Temperature in NLP? Retrieved July 5, 2024, from https://lukesalamone.github.io/posts/what-is-temperature/

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information processing & management*, *24*(5), 513–523.

Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, *44*(1.2), 206–226. https://doi.org/10.1147/rd.441.0206

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2020, February). DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. https://doi.org/10.48550/arXiv.1910.01108

Saravia, E. (2022, December). Prompt Engineering Guide. Retrieved July 5, 2024, from https://github.com/dair-ai/Prompt-Engineering-Guide

Schimanski, T., Bingler, J., Hyslop, C., Kraus, M., & Leippold, M. (2023, October). ClimateBERT-NetZero: Detecting and Assessing Net Zero and Reduction Targets. https://doi.org/10.48550/arXiv.2310.08096

Shen, Z., Zhang, R., Dell, M., Lee, B. C. G., Carlson, J., & Li, W. (2021, June). LayoutParser: A Unified Toolkit for Deep Learning Based Document Image Analysis. Retrieved April 10, 2024, from http://arxiv.org/abs/2103.15348

Sietsma, A. J., Callaghan, M., Biesbroek, R., Theokritoff, E., Thomas, A., ... Minx, J. C. (2022, February). Global Tracking of Climate Change Adaptation Policy Using Machine Learning: A Systematic Map Protocol. https://doi.org/10.21203/rs.3.pex-1836/v1

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., ... Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, *529*(7587), 484–489. https://doi.org/10.1038/nature16961

Singha, A., Cambronero, J., Gulwani, S., Le, V., & Parnin, C. (2023). Tabular Representation, Noisy Operators, and Impacts on Table Structure Understanding Tasks in LLMs. https://doi.org/10.48550/ARXIV.2310.10358

SLOCAT. (2023). *SLOCAT Transport and Climate Change Global Status Report* (tech. rep.). Retrieved July 26, 2024, from https://tcc-gsr.com/

Spokoyny, D., Laud, T., Corringham, T., & Berg-Kirkpatrick, T. (2023, July). Towards Answering Climate Questionnaires from Unstructured Climate Reports. https://doi.org/10.48550/arXiv.2301.04253

Stechemesser, A., Levermann, A., & Wenz, L. (2022). Temperature impacts on hate speech online: Evidence from four billion tweets. *NeurIPS 2022 Workshop on Tackling Climate Change with Machine Learning*. https://www.climatechange.ai/papers/neurips2022/41

Stede, M., & Patz, R. (2021). *The Climate Change Debate and Natural Language Processing* (tech. rep.).

Stein, R. A., Jaques, P. A., & Valiati, J. F. (2019). An analysis of hierarchical text classification using word embeddings. *Information Sciences*, *471*, 216–232. https://doi.org/10.1016/j.ins.2018.09.001

Strubell, E., Ganesh, A., & McCallum, A. (2019, June). Energy and Policy Considerations for Deep Learning in NLP. https://doi.org/10.48550/arXiv.1906.02243

Sui, Y., Zhou, M., Zhou, M., Han, S., & Zhang, D. (2024, July). Table Meets LLM: Can Large Language Models Understand Structured Table Data? A Benchmark and Empirical Study. https://doi.org/10.48550/arXiv.2305.13062

Sun, S., Yuan, R., Cao, Z., Li, W., & Liu, P. (2024, June). Prompt Chaining or Stepwise Prompt? Refinement in Text Summarization. https://doi.org/10.48550/arXiv.2406.00507

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning: An Introduction* (Second Edition). MIT Press.

Swarnakar, P., & Modi, A. (2021). NLP for Climate Policy: Creating a Knowledge Platform for Holistic and Effective Climate Action. *ArXiv*, *abs/2105.05621*.

Szomszor, M., & Adie, E. (2022). Overton: A bibliometric database of policy document citations. *Quantitative Science Studies*, *3*(3), 624–650. https://doi.org/10.1162/qss_a_00204

Thakur, A. S., Choudhary, K., Ramayapally, V. S., Vaidyanathan, S., & Hupkes, D. (2024, July). Judging the Judges: Evaluating Alignment and Vulnerabilities in LLMs-as-Judges. https://doi.org/10.48550/arXiv.2406.12624

Thulke, D., Gao, Y., Pelser, P., Brune, R., Jalota, R., ... Erasmus, D. (2024, January). ClimateGPT: Towards AI Synthesizing Interdisciplinary Research on Climate Change. https://doi.org/10.48550/arXiv.2401.09646

Torregrossa, F., Allesiardo, R., Claveau, V., Kooli, N., & Gravier, G. (2021). A survey on training and evaluation of word embeddings. *International Journal of Data Science and Analytics*, *11*(2), 85–103. https://doi.org/10.1007/s41060-021-00242-8

Tounsi, A., & Temimi, M. (2023). A systematic review of natural language processing applications for hydrometeorological hazards assessment. *Natural Hazards*, *116*(3), 2819–2870. https://doi.org/10.1007/S11069-023-05842-0/FIGURES/4

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., … Lample, G. (2023, February). LLaMA: Open and Efficient Foundation Language Models. https://doi.org/10.48550/arXiv.2302.13971

Trautmann, D. (2023, August). Large Language Model Prompt Chaining for Long Legal Document Classification. https://doi.org/10.48550/arXiv.2308.04138

Tyagi, N., & Bhushan, B. (2023). Demystifying the Role of Natural Language Processing (NLP) in Smart City Applications: Background, Motivation, Recent Advances, and Future Research Directions. *Wireless Personal Communications*, *130*(2), 857–908. https://doi.org/10.1007/S11277-023-10312-8/TABLES/8

UNDP. (2023). What are Long-term Climate Strategies, and how can they help us tackle climate change? Retrieved July 26, 2024, from https://climatepromise.undp.org/news-and-stories/long-term-climate-strategies-LTS-LTLEDS-climate-change

UNFCCC. (2016). The Paris Agreement. https://unfccc.int/sites/default/files/resource/parisagreement_publication.pdf

UNFCCC. (2018). Decision 4/CMA.1. Retrieved July 31, 2024, from https://unfccc.int/sites/default/files/resource/4-CMA.1_English.pdf

UNFCCC. (2024a). Conference of the Parties (COP) | UNFCCC. Retrieved July 25, 2024, from https://unfccc.int/process/bodies/supreme-bodies/conference-of-the-parties-cop

UNFCCC. (2024b). Nationally Determined Contributions (NDCs) | UNFCCC. Retrieved July 24, 2024, from https://unfccc.int/process-and-meetings/the-paris-agreement/nationally-determined-contributions-ndcs#NDC-Synthesis-Report

United Nations. (2024). Official Languages. Retrieved July 22, 2024, from https://www.un.org/en/our-work/official-languages

Vaghefi, S., Wang, Q., Muccione, V., Ni, J., Kraus, M., … Leippold, M. (2023, April). ChatClimate: Grounding Conversational AI in Climate Science. https://doi.org/10.2139/ssrn.4414628

van der Maaten, L., & Hinton, G. (2008). Visualizing Data using t-SNE. *Journal of Machine Learning Research*, *9*.

Varini, F. S., Boyd-Graber, J., Ciaramita, M., & Leippold, M. (2020, December). ClimaText: A Dataset for Climate Change Topic Detection. Retrieved July 18, 2024, from https://arxiv.org/abs/2012.00483v2

Varini, F. S., Boyd-Graber, J., Ciaramita, M., & Leippold, M. (2021, January). ClimaText: A Dataset for Climate Change Topic Detection. https://doi.org/10.48550/arXiv.2012.00483

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., … Polosukhin, I. (2017). Attention Is All You Need. http://arxiv.org/abs/1706.03762

Voigt, L., Nowok, R., Lehmann, E., & Tyrrell, M. (2024, March). Unveiling Vulnerabilities in Climate Policy: AI Tools for Insights into Climate Policy Inclusivity. Retrieved July 20, 2024, from https://www.blog-datalab.com/home/unveiling-vulnerabilities-in-climate-policy-ai-tools-for-insights-into-climate-policy-inclusivity/

Wang, G., Chillrud, L., & McKeown, K. (2021). Evidence based Automatic Fact-Checking for Climate Change Misinformation. https://doi.org/10.36190/2021.39

Wang, Z. (2023, May). Visualizing and Explaining Transformer Models. Retrieved August 14, 2024, from https://deepgram.com/learn/visualizing-and-explaining-transformer-models-from-the-ground-up

Wartmann, S., Shaikh, S., Moosmann, L., Urrutia, C., Essus, C., ... Fuertes, O. Z. (2023). *NDC Progress Indicators: A guidance for practitioners* (tech. rep.). Deutsche Gesellschaft für Internationale Zusammenarbeit (GIZ) GmbH. Retrieved July 25, 2024, from https://transparency-partnership.net/system/files/document/GIZ_NDC-Indicators-Paper_231031.pdf

Watkins, C. (1989). Learning From Delayed Rewards.

Webersinke, N., Kraus, M., Bingler, J., & Leippold, M. (2022, September). CLIMATEBERT: A Pretrained Language Model for Climate-Related Text. https://doi.org/10.2139/ssrn.4229146

Wei, J., Bosma, M., Zhao, V. Y., Guu, K., Yu, A. W., ... Le, Q. V. (2022, February). Finetuned Language Models Are Zero-Shot Learners. Retrieved July 1, 2024, from http://arxiv.org/abs/2109.01652

Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., ... Fedus, W. (2022, October). Emergent Abilities of Large Language Models. https://doi.org/10.48550/arXiv.2206.07682

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., ... Zhou, D. (2023, January). Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. Retrieved July 1, 2024, from http://arxiv.org/abs/2201.11903

Welfens, P. J. (2022). *Global Climate Change Policy: Analysis, Economic Efficiency Issues and International Cooperation*. Springer International Publishing. https://doi.org/10.1007/978-3-030-94594-7

Werbos, P. J. (1988). Generalization of backpropagation with application to a recurrent gas market model. https://doi.org/10.1016/0893-6080(88)90007-x

Wu, T., Terry, M., & Cai, C. J. (2022, March). AI Chains: Transparent and Controllable Human-AI Interaction by Chaining Large Language Model Prompts. https://doi.org/10.48550/arXiv.2110.01691

Zhang, B., Haddow, B., & Birch, A. (2023). Prompting Large Language Model for Machine Translation: A Case Study. https://doi.org/10.48550/ARXIV.2301.07069

Zhao, P., Zhang, H., Yu, Q., Wang, Z., Geng, Y., ... Cui, B. (2024, May). Retrieval-Augmented Generation for AI-Generated Content: A Survey. Retrieved May 6, 2024, from http://arxiv.org/abs/2402.19473

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., … Wen, J.-R. (2023, November). A Survey of Large Language Models. https://doi.org/10.48550/arXiv.2303.18223

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., … Fidler, S. (2015). Aligning Books and Movies: Towards Story-Like Visual Explanations by Watching Movies and Reading Books. *2015 IEEE International Conference on Computer Vision (ICCV)*, 19–27. https://doi.org/10.1109/ICCV.2015.11

Żółkowski, A., Krzyziński, M., Wilczyński, P., Giziński, S., Wiśnios, E., … Biecek, P. (2022). Climate Policy Tracker: Pipeline for automated analysis of public climate policies. http://arxiv.org/abs/2211.05852

Assertion

Ich versichere wahrheitsgemäß, die Arbeit selbstständig verfasst, alle benutzten Hilfsmittel vollständig und genau angegeben und alles kenntlich gemacht zu haben, was aus Arbeiten anderer unverändert oder mit Abänderungen entnommen wurde sowie die Satzung des KIT zur Sicherung guter wissenschaftlicher Praxis in der jeweils gültigen Fassung beachtet zu haben.

Karlsruhe, August 25, 2024 NICOLAS BECKER