Weak Supervision and Numerical Commonsense for Modeling Climate-related Text Documents

Daniel Spokoyny

CMU-LTI-24-006

April 25, 2024

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:

Emma Strubell, Carnegie Mellon University
John Blitzer, Google Research
William Cohen, Carnegie Mellon University & Google DeepMind
Taylor Berg-Kirkpatrick, University of California, San Diego (Chair)

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Dedicated to Yuri Spokoyny

Abstract

Large pretrained language models have shown remarkable versatility across a range of NLP tasks and domains, yet there has been limited attention on applying these models to the climate domain. An ever-growing body of unstructured climate textual documents contains crucial quantitative measurements on carbon emissions, reduction commitments (e.g. reduce CO2 emission per kilometre from passenger cars by 37.5%) and other climate-related information like policy goals. However, current NLP systems struggle to comprehend the semantic meaning of numbers and their units, and generalize poorly to concepts like policy goals in the climate domain. To address these issues, in the first part of the thesis we propose new model architectures that serve as a useful inductive bias for predicting numbers as continuous values, extend these to predict units and quantities jointly, and introduce a new task of predicting the correlation of multiple quantities in texts. In the second half of the thesis, we introduce a new benchmark for climate policy goal classification tasks and demonstrate that current climate-adapted NLP models perform no better than their general counterparts. To address this shortcoming, we utilize existing semi-structured climate questionnaires to train QA models with better transfer learning capabilities on climate documents. Finally, we tackle alignment of unstructured climate documents head-on with models we fine-tuned through weak-supervision, along with modern full-fledged LLMs via prompting and in-context learning. Together, this thesis attempts to lay a foundation for future work that combines numerical commonsense models for the climate domain, paving the way for novel applications in climate documents such as extracting critical climate measurements, mining correlative relationships between quantities, and using retrieval-augmentation for numerical query answering.

Acknowledgments

I'd first like to thank my undergraduate advisors and mentors who encouraged me to pursue interesting problems way beyond my comfort zone: Murat Karaorman, Fermin Moscoso del Prado Martin, Omer Egecioglu and William Wang. I would like to thank all of my collaborators Ben Zorn, Alex Polozov, Stefan Savage, Geoffrey Voelker, Chien-Sheng Wu, and Tom Corringham, as well as all of the students that I was able to work with during my research.

I extend my heartfelt thanks to my committee for their invaluable feedback on this thesis. In particular, I am deeply grateful to William for introducing me to John, where our internship project was the nucleation point for much of this work.

Research can be an incredibly difficult process and Taylor always set the bar high to push me to be a better researcher. Taylor gave me the freedom to pursue my curiosity and explore new research directions, often uncharted for both of us. Throughout my PhD, I have advised many students about graduate school, emphasizing the importance of choosing an advisor who is fundamentally a good person. This is what I did, and it has made all the difference.

I've had the pleasure of volunteering with many fantastic individuals through Climate Change AI. I'm incredibly privileged to have made so many good friends at both Carnegie Mellon and UCSD: Hieu, Xinyi, Michael, Vidhisha, Abhilasha, Danish, Dongyeop, Shrimai, Dheeraj, Shruti, Zhun, Paul, Sho, Dhruv, Artidoro, Sachin, Anjalie, Bodhi, Alex .*, Sumanth, Evan, David.

I have also been fortunate to have some of the most amazing lab members: Junxian, Zhao, Ivan, Yasaman, Maria, Harsh, Jack, Niloofar, Zachary, Kartik. The coffees, surf sessions, and cold beers with Nikita, Nikolai, Tyler, and Volkan have always been the essential pit stops that helped refuel me throughout my journey.

Thank you to my friends and family that have been a support system all throughout: Jeremy, Jared, Max, Natalie, Alex, Tina, Nana, Boris, May, and Elina. Finally, I'd like to thank my parents who brought my family to the US and provided the necessary foundation for my entire life's trajectory.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Globally, tens of thousands of climate reports have been generated by different *stakeholders* such as corporations, cities, states, and national governments either voluntarily or in response to regulatory pressure. These reports include climate assessments, climate legislation, agency reports, regulatory filings, and corporate ESG (Environmental, Social, and Governance) and CSR (Corporate Social Responsibility). The reports disclose vital information on greenhouse gas emissions and targets, carbon footprints, environmental commitments, and climate risks. This information is key to identifying and solving climate science problems, engineering climate solutions, making informed policy, and helping stakeholders make actionable plans to curb global greenhouse gas emissions. In addition to reporting in text-based documents, some entities also answer questionnaires or provide tabular data with qualitative and quantitative components. These reports are tens to hundreds of pages long and contain thousands of quantitative data points, along with numerous details on climate policy targets, initiatives, and impacts that are not yet standardized or accessible.

Climate researchers use these documents for a variety of purposes, we provide two such examples below:

1. Climate Watch Tracking: The World Resources Institute has built Climate Watch (an open data climate project) by manually labeling all climate commitments (e.g. use 80% renewable energy for steel production by 2030) in national climate action plans.[1] By doing so, Climate Watch allows climate researchers or policy makers to compare progress across countries and identify which areas need financing, education, policy changes or other resources. This transparency is critical as it has been shown to lead to higher levels of accountability and more effective policy making. We show an example of Climate Watch annotations in Figure 1.1.
2. Comparing Life Cycle Analysis: To better understand which products are more environmentally friendly, climate researchers may compare their life cycle analyses reports. A life cycle analysis is a process to calculate the total carbon footprint of a particular product from cradle to grave and may include all emissions from the resource extraction, production, product use, and waste disposal stages.

[1] https://www.climatewatchdata.org/

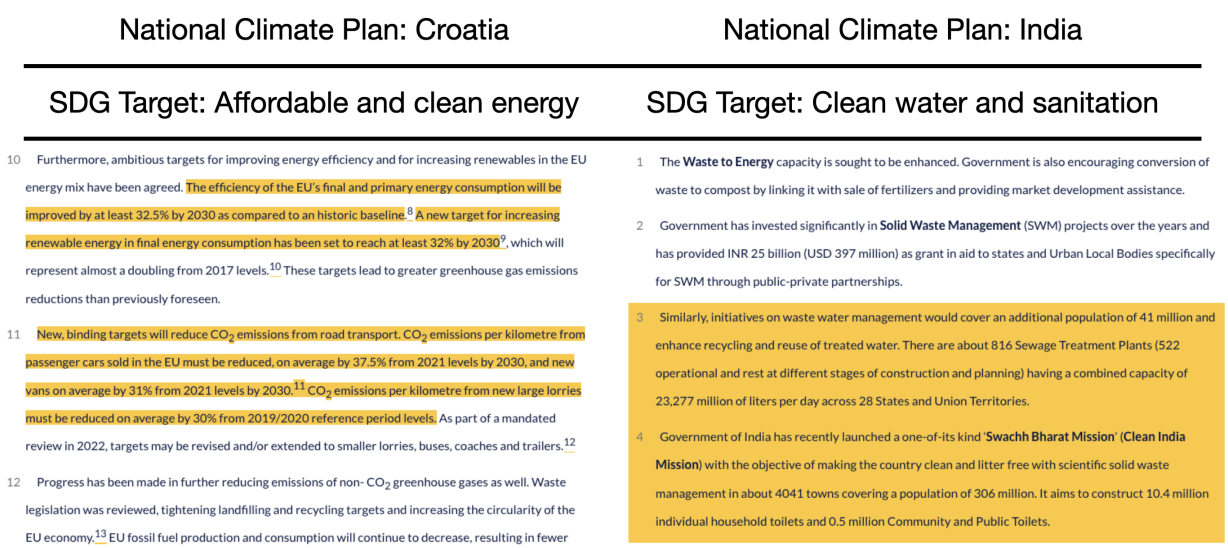


Figure 1.1: Here are two real examples of portions of Croatian and Indian national plans annotated with respect to "Clean Energy" & "Clean Water Sanitation" Labels as part of the Climate Watch project. The SDG target refers to the categorization of the label.

From these two examples we see that a real problem arises. With over hundreds of thousands of climate reports, the effort to manually extract information from these documents is simply not feasible. Yet these text documents contain the critical information required by the largest global coordination challenge in human history: reducing global climate emissions. From this it is clear that we need computational NLP methods that can process this data at scale to enable climate researchers to harness these climate documents to their full extent. These NLP systems should be capable of finding, extracting, and summarizing the various quantitative measurements on emissions, targets, impacts, and so on.

1.1 Challenges of Building NLP Systems for Climate Texts

There are many challenges that need to be overcome to build NLP systems that can process climate text data at scale. We will now shed light on three such challenges: *Lack of Standardization of Measurements*, *Lack of Standardization of Climate Document Types*, *Lack of Large Labeled Datasets*.

1.1.1 Standardization of Measurements

The International System of Units (SI) is the most widely used system of measurement. However, there is a lack of standardization in climate reporting requirements. Measurements expressed in SI units are from a purely dimensional analysis perspective always comparable. However, physical units alone are insufficient for us to describe and measure phenomena in the real world.

Semantic types (rainfall, carbon dioxide equivalents, humidity) describe what exactly is being measured. Although useful for global carbon disclosure, *total emissions of* CO_2 is not as

Ex#	*Samples of Real World Quantities*
1	These models, powered by a 140 kW / 190 Hp four-cylinder diesel engine, reduce fuel consumption by up to 0.3 litres per 100 km.
2	Starting in 2001, this five-year contract is for 5.25 million kWhs of wind-power per year.
3	In 2019, our intake of fresh water was 192 million cubic metres, compared with 199 million in 2018.
4	In the USA, the average fuel consumption for the model year 2019 was 35.7 mpg 2 for passenger cars (model year 2018: 35.1 mpg) and 29.8 mpg 2 for light trucks (model year 2018: 29.4 mpg). The average CO 2 emissions of both fleets is 33.7 mpg 2 or 167 g CO 2 / km 2 (model year 2018: 168 g CO 2 / km, internal BMW calculation). In China, average petrol consumption was 6.1 l / 100 km 3 in 2019 (2018: 6.2 l / 100 km), and the median CO 2 emissions were 14 4 g CO 2 / km 2 (2018: 147 g CO 2 / km).
5	Here, the BMW Group exceeded the previously announced target of having 500,000 electrified vehicles on the road since 2013 by selling around 504,000 units.
6	We aim to reduce the Net Carbon Footprint of the energy products we sell – expressed in grams of carbon dioxide (CO2) equivalent per megajoule consumed – by around 50% by 2050.
7	Tata Steel generates two key long-term, strategic performance indicators: specific energy consumption (Giga calorie / tonne of crude steel) and GHG intensity (tonne of CO 2 equivalent / tonne of crude steel).

Figure 1.2: Examples of measurements from sustainability reports. Example 4 uses footnotes in the same position as we would expect to find unit exponents and has many spacing issues in numbers and units. Examples 6 and 7 introduce new semantic unit types. Example 2 incorrectly pluralizes the unit symbol kWh"s". Example 3 and 5 use coreference to refer to units.

informative for corporations since changes in demand can drastically change emissions from year to year. For this reason, companies engineer their own key performance metrics with respect to climate change that capture their *efficiency* at a more fine-grained degree. For example, the multinational oil and gas company Shell calculates "grams of CO_2 equivalent per megajoule (gCO_2e/MJ) produced for each unit of energy delivered to, and used by, a consumer." However, companies prefer to use an even greater set of open class units that are customized to their own use cases and industries. For example the CSR report of automaker BMW calculates the emissions of CO_2e in proportion to a single vehicle produced.

In other types of climate studies, open class units are even more common. For example, the household appliances company Dyson, produced a study comparing their commercial hand drying units against a competing system and also against paper towels. To capture both the environmental economic impact as well as the functional use of the product, the study used a new functional unit they called *pair of dry hands*.

Finally, there are certain metrics that contain key financial information and are therefore only partially disclosed. For example, Google report their data center energy metric as the *noncomputing overhead energy use* divided by *IT equipment energy use* but only provide the resulting fraction and not the numerator or denominator values. Certain omissions can result from numeric fused-heads ("I am 19." can imply *years old*) which are commonplace in everyday English. The complexity of the types of measurements used in climate texts is further exacerbated by the use of non machine readable documents where existing PDF parsing tools struggle to extract text and strip invaluable information such as superscripts, formatting, capitalization, and spaces. See Figure 1.2 for real-world examples of measurements observed in climate documents.

Although we expect that companies will exhibit similar reporting within their own respective

industries, to capture the long-tail of types of quantities that appear in climate studies will be infeasible to manually codify. Since current models don't have symbolic reasoning required to handle even basic units, the semantic categories and open units combined with noisy text present a major hurdle for NLP models.

1.1.2 Lack of Standardization of Document Structure

The vast majority of climate reports are voluntary (although strongly encouraged). In the US it is still not required by law for public companies to disclose their carbon emissions. Further, there exists fear that oversweeping mandatory climate disclosures may result in companies going private where it becomes inconceivably more difficult to pass legislation for carbon reporting, while also hurting smaller business that may not be able to afford costly life cycle analysis. In contrast, climate researchers want as much data, as detailed, as standardized, comparable etc. For these reasons climate policy makers and organizations responsible for designing disclosures walk a fine line: encourage voluntary participation, help standardize reporting, and shine a light on new areas relevant to climate researchers. By asking well defined but open-ended questions policy makers are able to gauge how to proceed with more structure in future versions.

Further, standardization of climate reporting is a massive undertaking of experts (financial, accounting, policy, scientists, etc.) and is still evolving. It took over 80 years from the founding of the International Bureau of Weights and Measures (BIPM) in 1875 until the SI standard was born. The Climate Watch example we discussed above is in fact an example of an organization that's aim is to transform information in one type of climate report (national climate action plans) to a different climate framework (sustainable development goals). The difficulty of standardization is exacerbated by reports being in different languages and submitted at different points in time.

There are numerous different types of climate report types which contain often overlapping information with different levels of granularity and specificity. Since most climate reporting is voluntary many entities only have a small subset of applicable documents. This makes it difficult for NLP since there is not reliable formatting cues that are going to make this processing easy. Due to the lack of repeatable structure across this domain text understanding is going to be required.

1.1.3 Lack of Large Labeled Datasets

There are many resources available for training general-purpose NLP models, such as Wikipedia, Yelp data, Penn Treebank, GigaWord, and Internet crawls. Further, successful applications of NLP to domains such as biomedical or legal have been made possible by assembling specific resources, benchmarks, and annotated datasets. There are a handful of small existing datasets in the climate domain such as Twitter stance detection or Wikipedia climate relevance, which we discuss in more detail in Chapter 6. However, these datasets are only useful for evaluating model performance on small subsets of the climate domain. Further, the use of proprietary datasets for both pretraining as well as evaluation makes it difficult to effectively track NLP progress in the climate domain. For example, the Global Reporting Initiative (GRI), a non-governmental organization, once hosted over 63,000 corporate sustainability reports which are now inaccessible.[2]

[2]https://www.globalreporting.org/how-to-use-the-gri-standards/register-your-report/

One of the properties that makes climate documents unique is that they combine many different types of language, including technical, scientific, as well as policy language. They serve multiple different audiences, including policy makers, scientists, company shareholders, and the general public. For these reasons it is both challenging and expensive to label climate documents since it requires various types of climate expertise (which can still result in low inter-annotator agreement see Chapter 5).

1.2 Overarching Goals

In this thesis we will make strides to tackle portions of each of these challenges. We will not exhaustively address all aspects of the challenges presented above. For instance, our thesis will not touch on the orthographic variation of numbers, or on improving the extraction from PDF tools which are both important considerations but out of scope for this thesis. Instead we will focus on a few concrete capabilities that we want to imbue our NLP models with.

First we want to design models that treat numbers not as discrete tokens but as continuous values and support outputting distributions of quantities in text. Just as language models have been applied for auto-complete, these models capable of numerical guesstimation would too be useful. These systems can in turn be also used to detect anomalies in numbers be they from human error, data corruption, or natural phenomena. Finding such outliers could be helpful for a variety of climate applications such as: carbon accounting or aiding in the detection of greenwashing.

Second, we want to categorize and aggregate measurements according to their units. The models should learn that different units can be expressed while measuring the same natural phenomena e.g. *total rainfall* can be reported in either *inches per year* or *gallons per year* both implying a certain region with an area. When a unit is not mentioned we should infer it from the context. Our systems should be robust to such ambiguity and handle new units that we've never seen before.

Finally, we aim to combine these capabilities into a fundamentally new type of numerical commonsense QA system. The focus is not on symbolic reasoning or arithmetic operations, instead it's a correlative, quantitative, probabilistic framework that is learning numerical commonsense from large collections of text. This system should be able to answer questions like: "What is the average amount of carbon dioxide emitted by a 2000 lb car in the US?". It should learn that the weight of a vehicle correlates positively with the amount of carbon dioxide emitted. And it should be able to identify these salient quantities that may be related to each other.

To build systems capable of processing climate texts we propose to break down the problem into three main technical challenges: (1) modeling the distribution of numbers in context, (2) categorizing and aggregating measurements according to their units, and (3) information extraction from climate documents. We will discuss each of these challenges in detail in the following sections.

1.2.1 Goal 1: Modeling Distribution of Numbers in Context.

In natural language processing (NLP), numbers have been studied from different perspectives. (1) The use of NLP models for pushing the frontier of mathematical understanding. (2) Solving

algebraic expressions as a test bed for symbolic reasoning and as a method to harden existing NLP tasks to prevent models from overfitting to spurious correlations in the data. (3) Ingraining models with numerical commonsense knowledge for downstream applications such as information extraction, anomaly detection, and question answering. Our focus in this thesis is on the third perspective: building methods that can model numbers as real-valued continuous values and evaluating how well they can approximate the true distribution of numbers in texts.

Let us start with an example: "A 2019 study found that the average American produces #NUM# pounds of trash per day." What values of #NUM# are plausible? Is 5 pounds reasonable? What about 100 pounds or 0.1 pounds? This is task of *guesstimation* is one that humans are well-familiar with from a young age. We propose to build systems that can model the distribution of numbers from their surrounding text. Currently and for most of the past of NLP, when a number appears in data, it is simply treated as a special text token. Historically in NLP numbers were tokenized with simple regular expression patterns (ddd.d) that would capture only features of the shape of the number. Most tokenizers such as byte-pair encodings, treat numbers equal to the rest of the vocabulary of tokens. For these reasons in most NLP models, numbers are not treated as what they are: continuous values.

1.2.2 Goal 2: Categorization of Quantities

After being able to model the distribution of numbers in context, the next natural step is to be able to categorize numbers according to their *types*. The most evident example of this is the categorization of quantities to their corresponding physical units.

To illustrate an example consider a scenario where climate researcher studying coastal erosion may use such a system to extract data from California Coastal Commission emergency permits on feet of seawall, tons of riprap, and tons of sand applied for beach nourishment requested in response to extreme winter storm events. After studying the data she may hypothesize that the recent increase in flash floods are accelerating coastal erosion. To find such patterns she searches for data on rainfall measurements. However, since flash floods are caused by heavy rainfall, she is interested in the rate of rainfall, not the total amount rainfall. Further, she specifies that she is interested in rainfall measurements expressed in units of *millimeters per minute* as opposed to *inches per year*. From a dimensional analysis perspective *inches per year* and *inches per minute* are both units of *velocity* and are mathematically equivalent up to a scaling factor. However, from a semantic perspective of how we use these units these units behave differently. However, we tend to express measurements with units that can imply different semantic interpretations. Finally, she aggregates the data by year and sorts the results by heaviest rainfall rate.

1.2.3 Goal 3: Information Extraction for Climate Documents

In the past two goals we've highlighted specific goals: ability to infer a distribution over quantities in text, and ability to infer the unit types of measurements in text. These represent a new type of numerical commonsense, beyond basic declarative facts such as "a cat has four legs" and is distinct from symbolic reasoning in algebraic word problems. The numerical commonsense we are developing is a form of correlative, quantitative and probabilistic knowledge. It has unit types

that form a basis for reasoning which on its surface seem similar to symbolic reasoning, but are fundamentally different.

While improving numerical commonsense has many broad applications outside of understanding climate documents, the focus of our last goal is to build NLP models directly towards extracting information from climate documents. To do so we need 1) methods to improve in-domain performance without access to large scaled datasets 2) a benchmark to measure progress and 3) a practical system for climate researchers to use. We will work through the lack of large scaled datasets by leveraging the structure of climate questionnaires as weak-supervision to help bootstrap performance to build better in domain NLP models. To measure progress, we will construct new tasks and aggregate various climate text classification tasks into a single benchmark. These systems should be practical for use by climate researchers for a wide spectrum of text based tasks: ranging from searching for specific information in a large corpus or conducting exhaustive information extraction for an entire corpus. Further, these systems must also be highly dynamic, accommodating new tasks, taxonomies, and data sources as the field of climate science evolves. Towards this end, we will explore transfer learning as well as few-shot learning techniques to test how models adapt to changing data distributions and tasks.

1.3 Thesis Contributions and Proposed Work

Modeling Objective	**Sentence**	**Masked Tokens**
Masked Language Modeling	"A 50 foot redwood tree requires [MASK] gallons of water..."	"100"
Masked Numerical Modeling	"A 50 foot redwood tree requires [#NUM] gallons of water..."	100.0
Masked Measurement Modeling	"A 50 foot redwood tree requires [#NUM] [#UNIT] of water..."	(100.0, gallons)
Numerical Correlation	"A **50** foot redwood requires **100** gallons of water. . . "	Positive

Table 1.1: Example of the different modeling objectives we will consider in first section of this thesis.

Here we give a brief overview of the contributions of this thesis and how they relate to the three goals above. In Section 1, we present separate works on building numerical commonsense NLP models that target each of the goals above separately.

Since the unit is part of the text context, in principle it is possible that models could learn to make reasonable predictions for measurements from text alone. However, in Table 1.2 we show a simple failure mode that current models have at predicting numbers with differing units with traditional masked language modeling. We note that the number predictions vary significantly between different units that are common substitutes for each other (e.g. meters and feet) as well as units along with their corresponding acronyms (e.g. *feet* vs. *ft*). Although interpreting exactly what is happening under the hood is hard (neural models being black boxes), from this we can

Input: [UNIT]	m	km	ft	mi	yd	in	meters	kilometers	feet	miles	yards	inches	-
Output	200	10	200	2	100	1	200	20	20	2	50	3	-
Convertion factor	1	1000	0.3048	1609.34	0.9144	0.0254	1	1000	0.3048	1609.34	0.9144	0.0254	-
Metric Output	200.0	10000.0	60.96	3218.68	91.44	0.0254	200.0	20000.0	6.096	3218.68	45.72	0.0762	-
Mean (Metric Output)							-						3086.8 m
std (Metric Output)							-						5820 m

Table 1.2: Example outputs for **Alex Honnold climbed for [MASK] [UNIT]**.

hypothesize that perhaps models are failing to learn measurements altogether, and are instead relying on simpler bi-gram level statistics to coordinate the quantity and unit. This clear failure mode motivates us to consider a more structured approach to modeling measurements and in Table 1.1 we show examples of the three modeling objectives we consider in contrast to the standard masked language modeling task.

Below we provide a brief overview of the Chapters in the first Section on *Building Numerical Commonsense NLP*.

- **An Empirical Investigation of Contextualized Number Prediction.** We begin our investigations with *Goal 1* by studying the ability of bidirectional transformers (BERT) to predict numbers as continuous values over multiple domains of text with. We experiment with novel combinations of contextual encoders and output distributions over the real number line. Specifically, we introduce a suite of output distribution parameterizations that incorporate latent variables to add expressivity and better fit the natural distribution of numeric values in running text. We evaluate these models on two numeric datasets in the financial and scientific domain. Our findings show that output distributions that incorporate discrete latent variables and allow for multiple modes outperform simple flow-based counterparts on all datasets, yielding more accurate numerical prediction and anomaly detection. We also show evidence that our models can effectively utilize surrounding quantities in and benefit from general-purpose unsupervised pretraining.
- **Masked Measurement Prediction.** We extend our previous model to reason specifically about measurements by modeling dimensions, units, and quantities jointly. This work aligns with *Goal 2*, building NLP models that can robustly categorize quantities according to their physical units. We study pretrained transformer models (RoBERTa) and show that on linear probing analysis they significantly underperform jointly trained number-unit models, emphasizing the *existing gap of numerical commonsense* in NLP models and the benefits of our proposed modeling approach. We show that by explicitly modeling measurements jointly, models can infer units from the surrounding context with high fidelity, spot unit errors in Wikipedia articles, and outperform human annotators on measurement guesstimation.
- **Numerical Correlation in Text.** In our final work in this section we propose a new task of identifying a correlation relationship between two numbers in text, which is a component of *Goal 3*, building NLP models that can perform numerical commonsense question answering. Consider the final row in Table 1.1, as an example of this task where we wish for models to learn that taller trees *usually* require more water. To this end, we introduce a new dataset,

which contains over 2,000 Wikipedia sentences with two numbers and their correlation relationship labeled by human annotators. Using this dataset we show that our proposed techniques for numerically aware pretraining methods from earlier chapters lead to slight improvements on this task. However, these methods still underperform both larger models as well as the human baseline, posing a challenge for future work in this area.

In the second part of this thesis, we will delve into the specific challenges and opportunities of applying NLP to the climate change domain. Despite the abundance of climate-related text available, such as scientific papers, news articles, and social media posts, resources and tasks specifically tailored to the climate change domain are still limited. Given this hurdle, we have set our initial objective to construct classification models that can categorize sentences based on their climate-related topics instead. This will lay the foundation for our ultimate goal of categorizing climate quantities. As a result, the initial two chapters in Section 2 are focused on fulfilling *Goal 2*, without giving specific attention to the numerical data present in the sentences.

Below we provide a brief overview of the Chapters in the second Section on *Applying NLP models to Climate Domain*.

- **BERT Classification of Paris Agreement Climate Action Plans.** We use the document header structure of climate action plans to assign noisy policy-relevant labels such as mitigation, adaptation, energy, and land use to sentences. Transformers finetuned on this noisy labeled data provide only a slight improvement over simple heuristics and fall short of the consistency observed between human annotators.
- **Towards Answering Climate Questionnaires from Unstructured Climate Reports.** We introduce two new large-scale climate questionnaire datasets and use their existing structure to train self-supervised models. We conduct experiments to show that these models can learn to generalize to climate disclosures of different organizations types than seen during training. We then use these models to help align texts from unstructured climate documents to the semi-structured questionnaires in a human pilot study. Finally, to support further NLP research in the climate domain we introduce a benchmark of existing climate text classification datasets to better evaluate and compare existing general models with their domain-adapted counterparts.
- **Aligning Unstructured Paris Reports with SDG Framework** Finally, in Chapter 7 we construct a benchmark for evaluating alignment of unstructured climate reports according to the Sustainable Development Goals taxonomy. We reexamine our climate domain fine-tuned cross-encoder models and compare them against modern LLMs with prompting and in-context learning.

Chapter 2

An Empirical Investigation of Contextualized Number Prediction

- Daniel Spokoyny, Taylor Berg-Kirkpatrick, "An Empirical Investigation of Contextualized Number Prediction.", In Proceedings of the 2020 Conference of Empirical Methods in Natural Language Processing

2.1 Introduction

Pretraining large neural architectures (e.g. transformers [33, 89]) on vast amounts of unlabeled data has lead to great improvements on a variety of NLP tasks. Typically, such models are trained using a masked language modeling (MLM) objective and the resulting contextualized representations are finetuned for a particular downstream task like question answering or sentence classification [33, 60]. In this chapter, we focus on a related modeling paradigm, but a different task. Specifically, we investigate contextualized number prediction: predicting a real numeric value from its textual context using an MLM-style modeling objective. We conduct experiments on two specific variants: (1) *masked number prediction* (MNM), in which the goal is to predict the value of a masked number token in a sentence, and (2) *numerical anomaly detection* (NAD), with the goal of deciding whether a specific numeric value in a sentence is errorful or anomalous. In contrast with more standard MLM training setups, here we specifically care about the accuracy of the trained masked conditional distributions rather than the contextualized representations they induce. While successful models for these tasks are themselves useful in applications like typo correction and forgery detection [28], better models of numeracy are essential for further improving downstream tasks like question answering, numerical information extraction [80, 101] or numerical fact checking [118], as well as for processing number-heavy domains like financial news, technical specifications, and scientific articles. Further, systems that detect anomalous numbers in text have applications in practical domains – for example, medicine [117] – where identification of numerical entry errors is critical.

Our modeling approach to contextualized number prediction combines two lines of past work. First, following Chen et al. [28], we treat number prediction as a sentence-level MLM problem

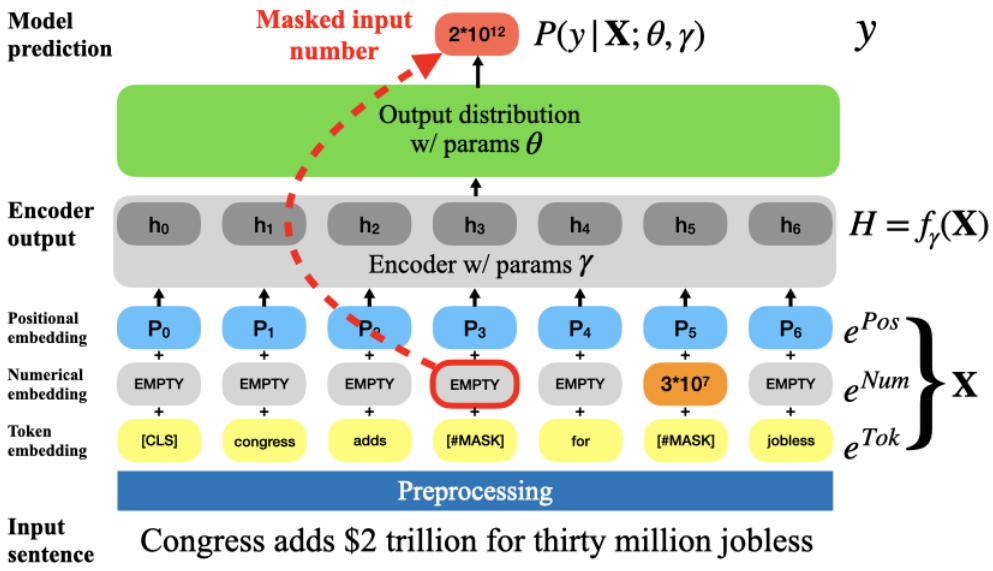


Figure 2.1: Outline of our model architecture consisting of a sentence representation $\boldsymbol{X}$ which is fed to the encoder with parameters γ and an output distribution over the real number line with parameters θ. In this example our masked numerical objective is to predict the masked out "2 trillion" quantity $\boldsymbol{Y}$. Note that our model is able to use a numerical embedding of the unmasked input $3 * 10^7$ value ("thirty million") as part of the context.

where only numerical quantities are masked. However, Chen et al. [28] focused on predicting the discrete exponent of masked numbers as a classification problem. In contrast, Spithourakis and Riedel [107] demonstrate the utility of predicting full numerical quantities in text, represented as real numbers, but do so in a language modeling framework, conditioned only on left context. Here, we propose a novel setup that combines full-context encoding (i.e. both left and right contexts) with real-valued output distributions for modeling numerical quantities in text. In Figure 2.1, we illustrate an example where we aim to predict "2 trillion" as a quantity on the real number line.

We expand upon past work by conducting a large scale empirical investigation that seeks to answer three questions: (1) Which encoding strategies yield more effective representations for numbers in surrounding context? (2) Which encoding architectures provide the best representations of surrounding context? (3) What are the most effective real-valued output distributions to model masked number quantities in text? To answer these questions, we propose a suite of novel real-valued output distributions that add flexibility through the use of learned transformation functions and discrete latent variables. We conduct experiments for both MNM and NAD tasks on two large datasets in different domains, combining output distributions with both recurrent and transformer-based encoder architectures, as well as different numeric token encoding schemes. Further, while Chen et al. [28] studied a specific type of NAD (detecting exaggerated numbers in financial comments), we examine several NAD variants with different types of synthetic anomalies that are found to arise in practice across different domains of data. Finally, we further compare results with a strong discriminative baseline.

2.2 Models

Our goal is to predict numbers in their textual contexts. The way we approach this is similar to masked language modeling (MLM), but instead of masking and predicting all token types, we only mask and predict tokens that represent numeric values. For example in Figure 2.1 we wish to predict that the value of the masked number `[#MASK]` should be $2 \times 10^{12} \in \mathbb{R}$ given the surrounding context.

For notational simplicity, we describe our model as predicting a single missing numeric value in a single sentence. However, like other MLMs (see section 2.4.3), during training we will mask and predict multiple numeric values simultaneously. Let $\boldsymbol{X}$ be a sentence consisting of N tokens where the kth token is a missing numerical value, $\boldsymbol{Y}$. The goal of our model is to predict the value of $\boldsymbol{Y}$ conditioned on $\boldsymbol{X}$. We will use common notation for from similar setups and simply treat the kth token in X as a masked numeric value, `[#MASK]`.

Our models $P_{\theta,\gamma}(y|\boldsymbol{X})$ consist of three main components: an input representation of the sentence, a contextual encoder with parameters γ which summarizes the sentence, and an output distribution with parameters θ over the real number line. In this section we will describe our strategies for numerical input representation, the two types of contextual encoders we use, along with different formulations of numerical output distributions.

2.2.1 Input Context Representation

We first describe the input representation for the textual context $\boldsymbol{X}$ that will be passed into our model's encoder. We let x_i represent the ith token in the input sequence. Like related MLMs that leverage transformers (which is one type of encoder we consider in experiments) we separate the representation of x_i into several types of embeddings. We include a positional embedding e^{POS} and a word-piece token embedding e^{TOK} like the original BERT. We also introduce our new numeric value embedding e^{NUM} to help us learn better numerical representations. Finally, as shown in Figure 2.1, the input representation for token x_i is the sum of these three H-dimensional embeddings.

If the token at position i represents a numerical quantity, we replace it with a special symbol `[#MASK]`, and represent its numerical value using e_i^{NUM}.[1] We use the extraction rules detailed in Section 2.3.1 to find the numbers in our input sequence. In the next section we will describe two strategies for numerical representation e^{NUM}.

Digit-RNN Embedding

The large range ($[1, 1e^{16}]$ in our data) of numerical values prevents them from being used directly as inputs to neural network models as this results in optimization problems due to the different scales of parameters. One strategy to learn embeddings of numerical values has been shown by Saxton et al. [104] which used character-based RNNs to perform arithmetic operations such as addition and multiplication. We conduct experiments with a similar strategy and represent each

[1] We exclude segment type embeddings since we do not perform next sentence prediction. We also found it helpful to use the zero vector as the numerical embedding for e_i^{NUM} if position i is not a quantity.

number in scientific notation (d.ddde+d) with 6 digits of precision as a string. We then use a digit-RNN to encode the string and use the last output as e^{NUM}.

Exponent Embedding

A simpler approach to represent numbers would be to explicitly learn embeddings for their magnitudes. Magnitudes have been shown to be a key component of the internal numerical representation of humans and animals [6, 32, 137]. We conduct experiments with an encoding scheme that learns embeddings for base-10 exponents.

2.2.2 Context Encoder

The encoder's goal is to summarize the surrounding text, along with other numbers that appear therein. We define $\boldsymbol{H} = f_\gamma(\boldsymbol{X})$ where the encoder f_γ is a function of the context $\boldsymbol{X}$, and $\boldsymbol{H}$ is the hidden representation of the encoder's last layer. Next, we describe two encoder architectures: a transformer and a recurrent approach.

Transformer Encoder

Transformer architectures pretrained on vast amounts of data have led to breakthroughs in textual representation learning [60, 69, 89, 143]. We use the 12-layer BERT-base architecture [33] with the implementation provided by Huggingface [139]. We use the original BERT's word-piece vocabulary with 30,000 tokens and add a new `[#MASK]` token.

BiGru Encoder

Previous methods focusing on the related task of predicting the order of magnitude of a missing number in text showed that RNNs were strong models for this task [28]. In our real-valued output task we use a bidirectional Gated Recurrent Unit (*BiGRU*), the best performing model from Chen et al. [28]. We use a one-layer BiGRU with a 64-dimensional hidden state and a dropout layer with a 0.3 dropout rate. We use the same pretrained word-piece embeddings from BERT as this allows us to directly compare the two encoders.

2.2.3 Real-valued Output Distributions

In early experiments, we observed that simple continuous distributions (e.g. Gaussian or Laplace) performed poorly. Since numbers can have ambiguous or underspecified units, and further, since numbers in text are heavy-tailed, asymmetric or multi-modal output distributions may be desirable. For this reason, we propose several more flexible output distributions, some which include learned transforms and others which include latent variables (both well-known methods for adding capacity to real-valued distributions), to parameterize $P(y|\boldsymbol{X})$.

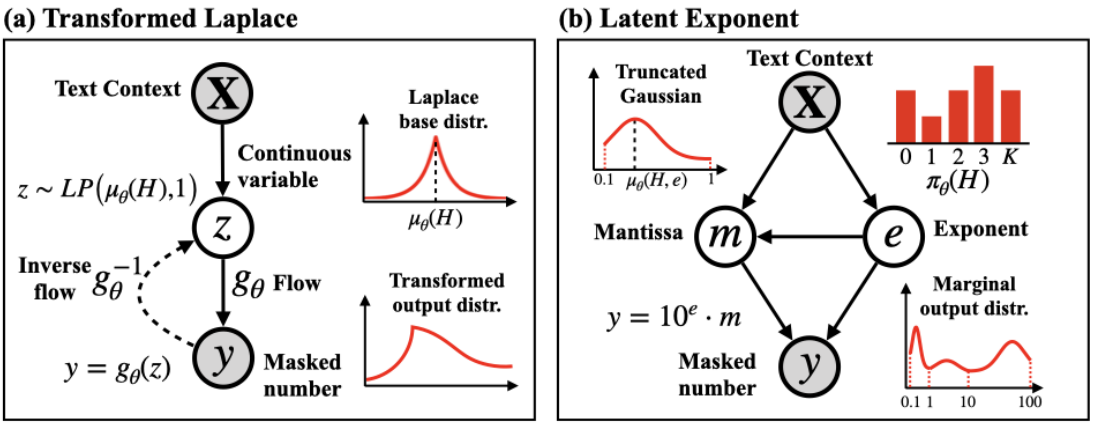


Figure 2.2: *Left (a):* We depict our *LogLP* and *FlowLP* graphical models along with the latent and output distributions. *Right (b):* Probabilistic graphical model of our latent *DExp* model.

Log Laplace

A common method for constructing expressive probability density functions is to pass a simple density through a transformation (e.g. a flow or invertible mapping function). As an initial example (and our first output distribution), we describe the log Laplace distribution as a type of flow. Since numbers in text are not distributed evenly on the number line due to a long tail of high magnitudes, a simple trick is to instead model the log of numeric values. If the base distribution is Laplace, this yields a log Laplace distribution, which we describe next as an exponential transformation.

In Figure 2.2, we illustrate our *LogLP* model with a continuous intermediate variable z, encoder f_γ, with exp as the transformation, g_θ , and consequently log as g_θ^{-1}. In equation 2.1 we show our generative process and training objective where both g_θ and g_θ^{-1} are deterministic functions with no parameters. We let $\mu_\theta(\boldsymbol{H})$ denote a single layer MLP that outputs the location parameter of the base Laplace distribution on z, which is transformed to produce the output variable, y. More precisely:

Generative Process:

$$z \sim Laplace(\mu_\theta(H), 1)$$

$$y = g_\theta(z) = \exp z$$

Training Objective:

$$g_\theta^{-1}(y) = \log y$$

$$\log P(y \mid \mathbf{X}) = -\left| g_\theta^{-1}(y) - \mu_\theta(H) \right| - C + \log J_{det}(y)$$

$$\log J_{det}(y) = \log \left| \frac{dg^{-1}}{dy}(y) \right| = \log \left| \frac{1}{y} \right| \tag{2.1}$$

Flow-transformed Laplace

The $\exp$ transformation may not be the ideal choice for our data. For this reason we consider a parameterized transform (flow) to add further capacity to the model. For our purposes, we are restricted to 1-dimensional transformations $g : \mathbb{R} \to \mathbb{R}$. Further, by restricting the class of functions, we ensure an efficient way of computing the log-derivative of the inverse flow, which allows us to efficiently compute likelihood. We conduct experiments with the simple parameterized flow described in Equation 2.2. We use a single layer MLP to independently predict each parameter *a,b,c* from $\boldsymbol{H}$, the output of $f_\gamma(\boldsymbol{X})$. We also scale the range of b, c to be between [0.1, 10] using a Sigmoid activation. Similarly to the *LogLP* setting, $\mu_\theta(\boldsymbol{H})$ is a single layer MLP which predicts the location parameter of the Laplace.

Generative Process:

$$z \sim Laplace(\mu_\theta(H), 1)$$

$$y = g_\theta(z) = \frac{\exp(\frac{z-a}{b})}{c}$$

Training Objective:

$$g_\theta^{-1}(y) = a + b \log cy$$

$$\log P(y \mid \mathbf{X}) = - \left| g_\theta^{-1}(y) - \mu_\theta(H) \right| - C + \log J_{det}(y)$$

$$\log J_{det}(y) = \log \left| \frac{dg^{-1}}{dy}(y) \right| = \log \left| \frac{b}{y} \right|$$

(2.2)

This parameterization of flow is designed to allow for (1) re-centering of the input variable (via parameter a), (2) re-scaling of the input (via parameter b), and (3) re-scaling of the output (via parameter c). Together, this leads to a family of inverse flows that are all log-shaped (i.e. they compress higher values), yet have some flexibility to change intercept and range.

Discrete Latent Exponent

While *FlowLP* adds flexibility over the *LogLP* model, both have the drawback of only being able to produce unimodal output distributions.[2] A well-established approach to parameterizing multi-modal densities is to use a mixture model. The mixture component is determined by a discrete latent variable in contrast with the continuous intermediate variable introduced in the flow-based models. In Figure 2.2 we show our *DExp* model where e represents an exponent sampled from a multinomial distribution, and m is the mantissa sampled from a truncated Gaussian.

Prior work has shown the effectiveness of cross-entropy losses on numerical training [28, 104]. For this reason we use a truncated Gaussian on the range of [0.1,1] to generate m, which effectively restricts back-propagation to a single mixture component for a given observation. The combination of exponent and mantissa prediction allows us to benefit from the effectiveness of cross-entropy losses, while at the same time getting more fine-grained signal from the mantissa

[2]In principle, more complicated flows could also have multiple modes – though they are more challenging to construct and optimize.

loss. In Equation 2.3 we show the *DExp* generative process and training objective. We let $\pi_\theta(\boldsymbol{H})$ denote a single layer MLP that outputs the multinomial parameters of $P(e|X)$. Similarly, we let $\mu_\theta(\boldsymbol{H}, e)$ denote a two layer MLP with a [.1,1] scaled Sigmoid that outputs the mean parameter of the mantissa normal distribution.

Generative Process:	**Training Objective:**
$e \sim Mult(\pi_\theta(H))$	$e^*(y) = \lfloor log_{10}(y) \rfloor$
$y \sim \mathcal{N}_{trunk[0.1,1]}(\mu_\theta(H, e), 0.05)$	$\log P(y \mid \mathbf{X}) = \log \Big[P\big(e = e^*(y)\big) \cdot \frac{1}{C} \exp\Big(-10 \Big(\frac{y}{10^{e^*(y)}} - \mu_\theta\big(H, e^*(y)\big) \Big)^2 \Big) \Big]$

(2.3)

Gaussian Mixture Model

Inspired by the best performing model from Spithourakis and Riedel [107] we also compare with a Gaussian mixture model (*GMM*). This model assumes that numbers are sampled from a weighted mixture of *K* independent Gaussians. During training the mixture from which a particular point was sampled from is not observed and so it is treated as a latent variable. We can optimize the marginal log-likelihood objective by summing over the K mixtures. In equation 2.4, *GMM* has *K* mixtures parameterized by K means and variances $\boldsymbol{\mu}, \boldsymbol{\sigma}$, respectively. Following Spithourakis and Riedel [107], we pre-train the parameters $\boldsymbol{\mu}, \boldsymbol{\sigma}$ on all the numbers in our training data $\mathcal{D}$ using EM. The means and variances are then fixed and our masked number prediction model only predicts mixture weights during training and inference. We let $\pi_\theta(\boldsymbol{H})$ denote a single layer MLP that outputs the mixture weights $P(e|X)$.

Generative Process:	**Training Objective:**
$\mu = [\mu_1, \mu_2, \dots \mu_k]; \sigma = [\sigma_1, \sigma_2, \dots \sigma_k]$ $e \sim Mult(\pi_\theta(H))$ $y \sim \mathcal{N}(\mu_e, \sigma_e)$	$\log P(y \mid \mathbf{X}) = \log \sum_{k=1}^{K} \Big[P(e = k) \cdot \frac{1}{C} \exp\Big(\frac{-(y - \mu_k)^2}{2\sigma_k^2} \Big) \Big]$

(2.4)

2.3 Data

Financial news Financial news documents are filled with many different ratios, quantities and percentages which make this domain an ideal test-bed for MNM. The **FinNews** is a collection of 306,065 financial news and blog articles from websites like Reuters[3]. We randomly break the documents into [train, valid, test] splits with [246065, 30000, 30000] respectively.

[3]www.kaggle.com/jeet2016/us-financial-news-articles

Since **FinNews** has many occurrences of dates and years, we also evaluate on a subset corpus, **FinNews-$** , to measure effectiveness at modeling only dollar quantities in text. **FinNews-$** is constructed exactly as **FinNews** , with the added requirement that the number is preceded by a dollar sign token ($). For all training and testing on **FinNews-$** , we only predict dollar values.

Academic papers Academic papers have diverse semantic quantities and measurements that make them an interesting challenge numeracy modeling. For this reason, we also use S2ORC, a newly constructed dataset of academic papers [72]. We use the first 24,000 full text articles, randomly splitting into [20000, 2000, 2000] [train, valid, test] splits. [4] We refer to this dataset as **Sci**. All three datasets follow the same preprocessing discussed below and summary statistics are provided in Table 2.1.

2.3.1 Preprocessing

Financial news, academic papers, and Wikipedia articles all have different style-guides that dictate how many digits of precision to use or whether certain quantities should be written out as words. While such stylistic queues might aid models in better predicting masked number *strings*, we are specifically focused on modeling actual numeric values for two reasons: (1) reduced dependence on stylistic features of the text domain leads to better generalization to new domains, and (2) the numerical value of a numeric token conveys its underlying meaning and provides a finer-grained learning signal. For example currencies are usually written as a number and magnitude like *$32 million* however, many quantities can be written out as cardinals *sixty thousand trucks*. We normalize our input numbers so that changing the style from *five* to *5* does not change our output predictions.

As exemplified in Figure 2.1, the aim of our approach is to incorporate both numbers as context and numbers as predictions (i.e. 2 trillion and thirty million in the example). For this reason, before tokenization we employ heuristics to combine numerals, cardinals and magnitudes into numerical values, whilst removing their string components. We also use heuristics to change ordinals into numbers. By following this normalization preprocessing procedure we get higher diversity of naturally occurring quantitative data and mitigate the bias towards some particular style guide.

For both **FinNews** and **Sci** we lowercase the text and ignore signs $(+, -)$, so all numbers are positive and restrict magnitudes to be in $[1, 1e^{16}]$. We discard sentences that do not have numbers or where the numbers are outside of our specified range. We also filter out sentences that have less than eight words and break up sentences longer than 50 words.[5] We do not use the special token *[SEP]* and all examples are truncated to a maximum length of 128 tokens.

[4] We also filter articles from only these categories {Geology, Medicine, Biology, Chemistry, Engineering, Physics, Computer science, Materials science, Economics, Business, Environmental science}.

[5] Sentences under eight words in length tended to be titles of articles with the date as the only numeric quantity.

	FinNews			FinNews-$			Sci		
	train	**valid**	**test**	**train**	**valid**	**test**	**train**	**valid**	**test**
#instances	522996	58095	64433	188286	22338	23281	360514	36523	36104
avg-length	102.5	108.3	108.9	115.2	115.4	116.1	125.6	126.4	126.5
%numbers	8.8	9.3	9.6	13.0	12.7	13.2	7.1	7.2	7.1
min	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
median 50	313.0	250.0	329.0	2016.0	2016.0	2016.0	9.0	8.0	9.0
median 75	3141.0	2558.0	3500.0	$\sim 10^4$	$\sim 10^4$	$\sim 10^4$	42.0	40.0	43.0
median 90	$\sim 10^6$	$\sim 10^6$	$\sim 10^6$	$\sim 10^7$	$\sim 10^7$	$\sim 10^7$	1959.0	1948.0	1972.1
max	$\sim 10^{15}$	$\sim 10^{14}$	$\sim 10^{15}$	$\sim 10^{15}$	$\sim 10^{14}$	$\sim 10^{15}$	$\sim 10^{15}$	$\sim 10^{14}$	$\sim 10^{15}$

Table 2.1: Statistics on our datasets. The top half of the table reveals the number of examples per data split, the average length of sentences, and the fraction of tokens that are numbers. The bottom half shows summary statistics for number values in both datasets.

2.4 Experiments

In this section we explain our experimental setup, starting with our evaluation metrics, implementation details, results, and ablation analyses. We use the following naming convention for models: we specify the encoder (*BiGRU*, *BERT*) first, followed by one of our four output distributions (*LogLP*, *FlowLP*, *DExp*, *GMM*).

2.4.1 Evaluation

For the MNM task on $\mathcal{D}_{\text{valid}}$ and $\mathcal{D}_{\text{test}}$ splits we randomly select a single number to mask out from the input and predict. We let $\hat{y}$ denote the model's $\arg\max$ prediction from $P(y|\boldsymbol{X})$ and $\boldsymbol{Y}$ as the actual observed number. In equation 3.2 and 2.6 we show how we calculate log-MAE (*LMAE*) and exponent accuracy (*E-Acc*), both of which use log base 10.

$$\textit{LMAE} = \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{\mathcal{D}_{\text{test}}} |\log \boldsymbol{Y} - \log \hat{y}| \tag{2.5}$$

$$\textit{E-Acc} = \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{\mathcal{D}_{\text{test}}} \mathbb{1}[\lfloor \log \boldsymbol{Y} \rfloor = \lfloor \log \hat{y} \rfloor] \tag{2.6}$$

2.4.2 Numerical Anomaly Detection

Both *LMAE* and *E-Acc* metrics test the model's argmax prediction and not the entire $P(y|\boldsymbol{X})$ distribution. We next consider the NAD task where our models need to discern the true number versus some anomaly. We let $\tilde{y}$ denote an anomaly and describe two different ways, [*string, random*], we construct an anomalous example. For *string* we use the true $\boldsymbol{Y}$ and randomly perform one of three operations [*add, del, swap*]: inserting a new digit, deleting an existing digit, and swapping the first two digits respectively. For *random*, we randomly sample a number from the training data $\mathcal{D}$ as our anomaly. We choose these string functions as they constitute a large part

of numerical entry errors [117, 138]. Further, *random* mimics a copy-paste error. We report the AUC of a ROC curve for both types as random-anomaly (*R-AUC*) and string-anomaly (*S-AUC*) respectively, using the model's output density to rank the true value against the anomaly.

2.4.3 Implementation Details

We train all models with stochastic gradient descent using a batch-size of 32 for 10 epochs. We use early stopping with a patience of three on the validation loss. For pretrained *BERT* encoder experiments, we use two learning rates $\{3e^{-5}, 1e^{-2}\}$ for all pretrained parameters and newly added parameters respectively. For all non-pretrained *BERT* experiments and all *BiGRU* encoders we use a single learning rate of $2e^{-2}$.

Devlin et al. [33] propose a two step process to generate masked tokens. First, select tokens for masking with an independent probability of 15%. Second, for a selected token: With 80% probability replace it with a `[MASK]`, 10% replace it with a random token, and 10% leave it unchanged. Since there are fewer numbers than text tokens, we use a higher probability of 50% for selection. We follow a similar strategy for masking numbers: 80% of the time masking out the number, 10% of the time randomly substituting it with a number from train, and 10% of the time leaving it unchanged.

We also consider a fully discriminative baseline trained to predict real vs. fake numbers with binary cross entropy loss. The negative numerical samples are randomly drawn from training set numbers to match exactly the random-anomaly task. During training each positive datum has one negative example and is trained in the same batch-wise fashion. When this model uses exponent embeddings for output numbers, emb_{exp} , we can also calculate the exponent accuracy by selecting the exponent embedding with highest model score as a predicted value. We include this approach in experiments as a non-probabilistic alternative to our four output distributions.

2.4.4 Results

We ran all combinations of encoders and output distributions using input exponent embeddings on **FinNews** and show the results in Table 2.2. We train the *GMM* model with four different settings of $K \in \{31, 63, 127, 255\}$ and report results for the highest-performing setting.

Comparing the two encoders, we find that *BERT* results in stronger performance across all metrics and all output distributions. Although both settings share the same pretrained embedding layers, the pretrained transformer architecture has higher capacity and is able to extract more relevant numerical information for both MNM and NAD.

We find that the parameterized *FlowLP* model was generally better across all metrics under both encoders compared to the *LogLP* model. With the weaker *BiGRU* encoder, the *LogLP* model's *S-AUC* is only 0.04 better than random guessing.

The *DExp* model was the best performing output distribution across all metrics and both encoders, yielding on average 10% higher *E-Acc* and a gain of 0.13 on *AUC*. This means that *DExp* had the best overall fit in terms of the predicted mode ($\arg\max$) as well as the overall density $P(y|\boldsymbol{X})$.

In contrast, *GMM* , which is also a discrete latent variable model capable of outputting a multimodal distribution, underperformed across all metrics. There was little effect from adjusting

Model	LMAE↓	*E-Acc* ↑	r-AUC↑	s-AUC↑
Train-Mean	7.69	1.03	-	-
Train-Median	1.88	5.52	-	-
BiGRU-Disc	-	55.8	0.756	0.646
BiGRU-LogLP	0.671	58.8	0.675	0.548
BiGRU-FlowLP	0.622	61.8	0.694	0.591
BiGRU-DExp	0.576	71.5	0.843	0.821
BERT-Disc	-	62.7	0.762	0.656
BERT-GMM K=255	1.18	21.3	0.585	0.440
BERT-LogLP	0.5666	64.9	0.686	0.557
BERT-FlowLP	0.5732	65.5	0.717	0.609
BERT-DExp	**0.500**	**74.6**	**0.861**	**0.828**

Table 2.2: Results on **FinNews** where all models use input exponent embeddings emb_{exp} and all *BERT* encoders are pretrained. We also include the mean and median number from training $\mathcal{D}$ as simple baselines.

the number of mixture components, with slight improvements using more mixtures. One possible reason for the *GMM* model's worse performance is that the mixtures are fit and fixed before training without any of the surrounding textual information. Quantities such as dates and years have many textual clues, but the model's initial clustering may group them together with other quantities. We also found that, empirically, optimization for this model was somewhat unstable.

Finally the *Disc* baseline was the second best performing model on NAD , though on MNM it showed worse *E-Acc* than *LogLP* and *FlowLP* models. This baseline benefited from being directly trained for NAD , which may explain it's under-performance on MNM metrics. Due to the comparatively worse performance of both the *BiGRU* encoder and the *GMM* output distribution, we exclude them from the remainder of our experiments.

2.4.5 Ablations

Ablations on Numerical Embedding We select our best performing model, *BERT-DExp*, and ablate the numerical input representation on **FinNews**. We compare using emb_{dig}, emb_{exp} , and a version of *ExpBert* which has no numerical input representation. The top half of Table 2.3 displays the results. We see that emb_{dig} and emb_{exp} perform equally well. Using no input number embeddings reduces performance by 8% on *E-Acc* and 0.03 *AUC* on both anomaly metrics. We also see that there is no benefit from combining both of these input representations, which implies that the model is able to extract similar information from each.

Ablations One-vs-All To measure our model's effectiveness at using the other numbers in the input we construct an ablated evaluation *All* , where all input numbers are masked out.[6] In Table 2.3 we see that all models that have a numerical embedding suffer a performance drop of around 12% *E-Acc* and an increase of 0.4 on *LMAE*. This suggests that the model is in fact

[6]To make comparisons exact, every test example has at least 2 numerical values so that we can perform this ablation.

Ablation Type	LMAE↓	*E-Acc* ↑	r-AUC↑	s-AUC↑	all-LMAE↓	all-*E-Acc* ↑
Numerical Input Embedding						
BERT-DExp (All #'s Masked)	0.656	66.5	0.831	0.809	0.656	66.5
BERT-DExp + emb_{exp}	0.500	74.6	0.861	0.828	0.888	62.2
BERT-DExp + emb_{dig}	0.506	74.4	0.858	0.826	0.920	62.1
BERT-DExp + emb_{exp} + emb_{dig}	0.498	74.9	0.861	0.828	0.899	62.3
No Pretraining						
BERT-DExp + emb_{exp}	0.615	68.8	0.840	0.810	0.889	60.6
BERT-FlowLP + emb_{exp}	0.769	57.9	0.670	0.563	0.861	54.4
BERT-Disc + emb_{exp}	-	26.9	0.632	0.599	-	-
BERT-LogLP + emb_{exp}	0.630	63.2	0.678	0.550	0.850	57.1

Table 2.3: Ablation on **FinNews** dataset. The top half of the table shows the effect of the numerical input representation. The bottom half shows performance for models trained from scratch, without leveraging pretrained BERT parameters.

using the other quantities for its predictions. We also find that the model with no input number embeddings does better on the *All* setting since it was effectively trained with fully masked input numbers.

Ablations on Pretraining In the bottom half of Table 2.3, we compare the effect of starting from a pretrained transformer versus training from scratch. We see that training from scratch hurts all models by around 6% on *E-Acc* and 0.02 on *R-AUC*. We also note that *BERT-LogLP* seems least affected, dropping only 1% on *E-Acc*.

Modeling Additional Domains In this section we explore how different models behave on the alternative domain of academic papers, and how modeling is affected by focusing only dollar quantities in financial news. In Table 2.4, we show results for pretrained BERT encoder models with input exponent embeddings, trained and evaluated on **Sci** and **FinNews-$** datasets.

On the **Sci** data, the generative models have similar performance on *LMAE* and *E-Acc* . We further find that *BERT-DExp* is still the best performing model across most metrics on both **Sci** and **FinNews-$** data. The *BERT-Disc* baseline, which is directly trained to predict anomalies, is consistently the second best across all datasets on NAD. Finally, we find that the **FinNews-$** is the most challenging of the three datasets, with *BERT-DExp* dropping on *E-Acc* by 20% compared to **FinNews** data. This supports our initial reasoning that the distribution of dollar amounts is more difficult to characterize than other quantities, such as dates, which tend to cluster to smaller ranges.

2.5 Related Work

Math & Algebraic Word Problems: There is a wide literature on using machine learning to solve algebraic word problems [67, 97, 144], building novel neural modules to directly learn numerical operations [78, 120] and solving a variety of challenging mathematical problems

Model	FinNews-$				Sci			
	LMAE↓	***E-Acc* ↑**	**r-AUC↑**	**s-AUC↑**	**LMAE↓**	***E-Acc* ↑**	**r-AUC↑**	**s-AUC↑**
BERT-Disc	-	46.9	0.828	0.588	-	68.8	0.722	0.657
BERT-LogLP	1.04	43.6	0.641	0.528	**0.374**	78.2	0.624	0.609
BERT-DExp	**0.91**	**56.9**	**0.867**	**0.678**	0.385	**81.0**	**0.786**	**0.836**
BERT-FlowLP	1.11	39.3	0.538	0.518	**0.374**	77.6	0.658	0.672

Table 2.4: Results on **FinNews-$** and **Sci** where all models use input exponent embeddings emb_{exp} and all *BERT* encoders are pretrained.

[59, 61, 104]. In these tasks, numbers can be treated as symbolic variables and computation based on these values leverages a latent tree of arithmetic operations. This differs from our task setting since there is no "true" latent computation that generates all the quantities in our text given the available context.

Numerical Question Answering The DROP dataset [36] is a dataset that requires performing discrete numerical reasoning within a traditional question answering framework. Andor et al. [5] treat DROP as a supervised classification problem, while recent work by Geva et al. [43] show how synthetic mathematical training data can build better numerical representations for DROP. Unlike work on DROP, our primary focus is on the task of contextualized number prediction and numerical anomaly detection in text, which involve correlative predictions based on lexical context rather than concrete computation.

String Embeddings Recently, word and token embeddings have been analyzed to see if they record numerical properties (for example, magnitude or sorting order) [84, 130]. This work finds evidence that common embedding approaches are unable to generalize to large numeric ranges, but that character-based embeddings fare better than the rest. However, this line of work also found mixed results on overall numeracy of existing embedding methods and further investigation is required.

Numerical Prediction Spithourakis and Riedel [107] trained left-to-right language models for modeling quantities in text as tokens, digits, and real numbers using a *GMM*. Our empirical investigation focuses on MNM and considers both left and right contexts of numbers, along with a broader class of generative output distributions. Chen et al. [28] predict magnitudes of numbers in text and also consider a type of NAD to detect numerical exaggerations on financial data. However, this modeling approach is restricted: it can only distinguish anomalies that result in a change of exponent. In contrast, our real-valued distributions allow us to focus on a broader suite of harder anomaly detection tasks, such as random substitutions and string input error.

2.6 Conclusion

In this work we carried out a large scale empirical investigation of masked number prediction and numerical anomaly detection in text. We showed that using the base-10 exponent as a discrete

latent variable outperformed all other competitive models. Specifically, we found that learning the exponent representation using pretrained transformers that can incorporate left and right contexts, combined with discrete latent variable output distributions, results is the most effective way to model masked number quantities in text. Future work might explore combining more expressive flows with discrete latent variables.

Chapter 3

Masked Measurement Prediction

- Daniel Spokoyny and Ivan Lee and Zhao Jin and Taylor Berg-Kirkpatrick, "Masked Measurement Prediction: Learning to Jointly Predict Quantities and Units from Textual Context.", In Proceedings of Findings NAACL 2022

3.1 Introduction

In this Chapter, we focus on a special subset of quantities: measurements. First, as an example of masked number prediction (*MNP*), given the sentence *"Cats have [#NUM] paws."* a model learns to predict the number *4*. While appropriate for numerical commonsense, *MNP* is deficient when it is used to predict measurements. *Measurements*, such as *2 meters* or *13.2 square miles*, are a special class of particularly common numbers in text that have a well-defined and typed system of *units*. Given a simple question: *"How long did Alex Honnold climb for?"*, a single number alone is an insufficient answer since it is meaningless without the unit. Answers like *1000 meters* or *4 hours* could both suffice.

Current *MNP* systems do not jointly reason about numbers *with* units. It is reasonable to expect that pretrained models like BERT could leverage information of units directly as text without any special treatment. However, as you recall from the preliminary experiments in the introduction, we find that models yield poor numerical abilities. Furthermore, including units as text directly raise more questions: should we evaluate using all units (*meters, feet, inches*)? Should we equally weight across the units? Current models have no opinion about which unit is appropriate because they are not required to make unit predictions during training. Together, this indicates that current training objectives do not capture sufficient representations of measurements and that a direct application of *MNP* to evaluate numeracy of measurements is ill-suited.

To address these shortcomings, we propose the more challenging task of Masked Measurement Prediction (*MMP*) along with a new model. In this task, a model must reconstruct both the number together with the correct unit. In Figure 3.1 we show how in a *MMP* setting our model generates a dimension ("Length"), a number in metric log-space ("3.00"), the unit ("feet") and then uses the conversion factor ("3.28") to deterministically output the full measurement ("3280 feet"). This example illustrates a key distinction in that our model is flexible and can generate *non-metric*

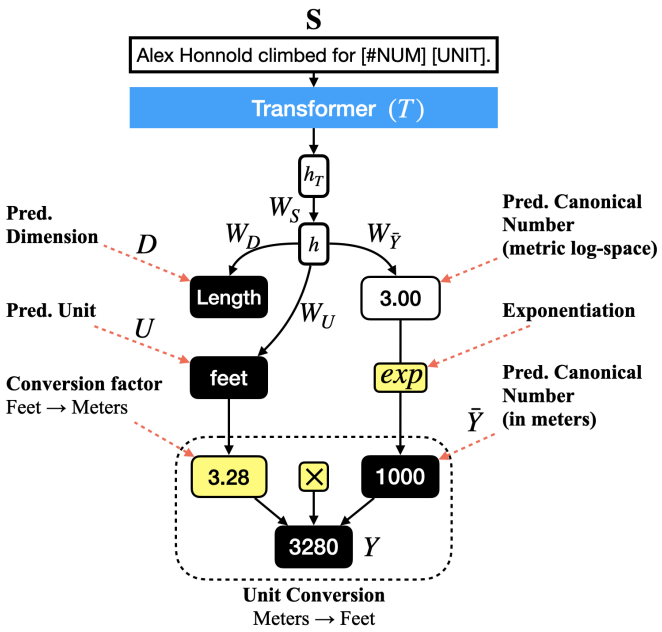


Figure 3.1: We present the Masked Measurement Prediction (*MMP*) task where the model predicts the dimension, unit and real-valued number. We also show the model architecture of **Ge**nerative **M**asked **M**easurement model (*GeMM*), the model we propose to perform *MMP*. We display the fixed operations used during unit conversion in yellow. In black, we show the different components of the model's prediction.

measurements (feet) but evaluates numerical prediction in canonical units (meters).[1]

MMP is useful for two reasons: 1) as a way to *train* models to give them better numeracy 2) as a new kind of *evaluation* that allows for a much more fine-grained analysis of reasoning over numerical quantities. The task of measurement estimation decouples the different aspects of numeracy allowing for a more interpretable and thorough analysis of numerical reasoning. We introduce a new evaluation benchmark for *MMP* based on Wiki-Convert (*WiCo*) [115], a large scale dataset of English Wikipedia sentences with ground truth measurement annotations. We compare the performance of our models on their ability to accurately predict the dimension, unit, and value of a measurement. We employ a large pretrained transformer model as our textual encoder and examine the performance of different discriminative, generative, and latent variable models along with several ablations. Our contributions are as follows:

- We introduce a novel challenging task *MMP* for pretraining and evaluating numeracy.
- We show that linear probing of existing pretrained models on MMP *significantly underperforms* fully finetuned models.
- We train a model that reasons jointly about numbers and units which predicts numbers 8.1 times more accurately than the probed pretrained models.
- We find our best performing generative model outperforms human annotators on two

[1]Our metric of choice described in Equation 3.2 is invariant to the specific choice of canonical unit i.e., *log-mae* in meters is equal to *log-mae* in feet.

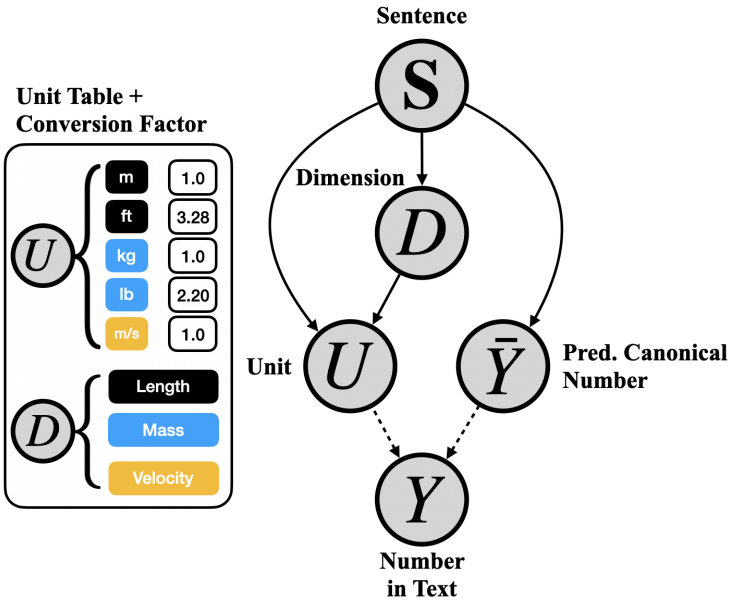


Figure 3.2: *GeMM* as a graphical model. The broken arrows represent a deterministic unit conversion. Examples of unit values and their corresponding dimension values are also shown.

evaluations, achieving 7.4-7.8% better dimension accuracy and 33.5-39.9% better unit accuracy. Furthermore, this model predicts a number **closer** to ground truth than our annotators 66.2-78.8% of the time.

Furthermore there are numerous applications of better measurement prediction and unit reconstruction such as in table to text generation [83], answering numerical queries [47, 103] or for improving e-commerce product search[8]. We hope that Masked Measurement Prediction becomes a standard benchmarking tool from which we can gain insight how to best incorporate new numeracy modeling techniques as well as evaluate existing models.

3.2 Models

3.2.1 Background + Notation

The International System of Units (SI) defines seven *fundamental dimensions* (Length, Time, Mass, etc.) and seven corresponding *base SI units* (meters, seconds, kilograms, etc.). The SI system is the most widely adopted measurement standard and is used internationally in domains such as commerce, finance, logistics, and science. We designate $\mathcal{D}$ to be the set of composite dimensions obtained from (and including) the *fundamental dimensions*. Values of $\mathcal{D}$ include velocity and power. We let $\mathcal{U}$ be the set of all units: the various ways to describe dimensions. For example, units of Length include meters and miles. Each training example consists of a real number y, a dimension $d \in \mathcal{D}$, a unit $u \in \mathcal{U}$, and the remainder of the sentence $\boldsymbol{S}$. In *MMP*, our task is to predict y, d, and u given only $\boldsymbol{S}$. In the next sections we describe our generative model designed for *MMP* followed by the ablations we consider.

3.2.2 Model

Measurements have complex semantic meanings, shaped by many standards, particular instruments, and natural world phenomena. Consider a text concerning rainfall. From a dimensional analysis perspective, the units *inches per year (in/y)* and *meters per second (m/s)* share the same dimension *velocity*. However, mentioning *in/y* usually implies that the text is discussing total rainfall in a region. Likewise, the use of *m/s* suggests that the text is examining the speed of falling rain droplets. To capture this complexity, we consider a generative model that learns the joint distribution of the number, dimension, and unit.

We now describe the generative process of our full model. To start, conditioned on $\boldsymbol{S}$, our model samples a discrete dimension variable $\boldsymbol{D}$. Then conditioned on the sampled dimension, our model samples a discrete unit variable $\boldsymbol{U}$ compatible with the dimension. For example, conditioned on the dimension *velocity* our model will output a distribution over the units of velocity such as *[miles per hour; meters per second, inches per year]* as opposed to all of $\mathcal{U}$. We then separately predict a distribution on the canonicalized measurement, $\bar{\boldsymbol{Y}}$, which is the numerical quantity represented in a base canonical (metric) unit like meters. During inference time, we use the highest scoring dimension and unit and choose the proper conversion factor to deterministically produce the final number y represented in the predicted unit. We refer to this **Ge**nerative **M**asked **M**easurement model as *GeMM*, where the joint $p(\boldsymbol{D}, \boldsymbol{Y}, \boldsymbol{U}|\boldsymbol{S})$ is given by the following equation:

$$p(\boldsymbol{D}|\boldsymbol{S}) \times p(\boldsymbol{U}|\boldsymbol{D}, \boldsymbol{S}) \times p(\boldsymbol{Y}|\boldsymbol{S})$$

We show the graphical model of *GeMM* in Figure 3.2. We also consider, *GeMM* U→Y, a slight variant where we have a direct dependence between the unit and number prediction with a joint equal to:

$$p(\boldsymbol{D}|\boldsymbol{S}) \times p(\boldsymbol{U}|\boldsymbol{D}, \boldsymbol{S}) \times p(\boldsymbol{Y}|\boldsymbol{U}, \boldsymbol{S})$$

3.2.3 Discrete Latent Dimension Model

We also consider an unsupervised generative model which treats the dimension as a discrete latent variable. We use the same number of dimension classes $|\mathcal{D}|$ and train to maximize the log-likelihood of the observed $\boldsymbol{Y}$. We refer to this model as *Lat-Dim* and is characterized by:

$$p(\boldsymbol{Y}|\boldsymbol{S}) = \sum_{\boldsymbol{D}} p(\boldsymbol{D}|\boldsymbol{S}) \times p(\boldsymbol{Y}|\boldsymbol{D}, \boldsymbol{S})$$

To evaluate this model we build a contingency matrix of the predicted classes and using a linear solver find the best mapping between our predicted and true dimensions. We can then apply this mapping to the model predictions and calculate classification metrics for dimension prediction.

3.2.4 Model Ablations

We also consider several model ablations of *GeMM*. Our first ablation is *GeMM* -Y -U which models $p(\boldsymbol{D}|\boldsymbol{S})$. The second, *GeMM* -Y, learns the distribution $p(\boldsymbol{U}, \boldsymbol{D}|\boldsymbol{S}) = p(\boldsymbol{D}|\boldsymbol{S}) \times p(\boldsymbol{U}|\boldsymbol{D}, \boldsymbol{S})$. The third, *GeMM* -U, models $p(\boldsymbol{Y}, \boldsymbol{D}|\boldsymbol{S}) = p(\boldsymbol{D}|\boldsymbol{S}) \times p(\boldsymbol{Y}|\boldsymbol{D}, \boldsymbol{S})$. Our final ablation is *GeMM* -U -D which learns $P(\boldsymbol{Y}|\boldsymbol{S})$ directly.

Split	Examples	Max #	Min #
All	919,237	5.5E+36	1E-06
Train	728,629	5.5E+36	1E-06
Val	91,110	4.4E+14	1.2E-06
Test	91,092	1.6E+21	1.8E-06

Table 3.1: Summary statistics for Wiki-Convert. The median number of characters and tokens per example is 106 and 33, respectively.

3.2.5 Model Architectures

For our textual encoder, we use the Huggingface Transformers [69, 140] implementation of RoBERTa, a pretrained 12-layer transformer. We refer to this text encoder as $\boldsymbol{T}$ such that given a sentence $\boldsymbol{S}$, our model outputs a 768-dimensional vector $\boldsymbol{h_T}$. We use a single linear layer, $W_S \in \mathbb{R}^{768 \times M}$, to project $\boldsymbol{h_T}$ to $\boldsymbol{h}$ and treat the dimension M as a hyper-parameter. To form a distribution over the real number line $\mathbb{R}$ we use a *Log-Laplace* model, a competitive model used in the numeracy literature [108, 115, 145]. This is equivalent to L_1 regression in log-space and yields the following loss function where $\boldsymbol{Y}$ and $\boldsymbol{Y}^*$ are predicted and ground truth numbers, respectively:

$$\log P(\boldsymbol{Y}|\boldsymbol{S}) = |\log \boldsymbol{Y}^* - \log \boldsymbol{Y}| + \log \left| \frac{1}{\boldsymbol{Y}} \right| \tag{3.1}$$

As shown in Figure 3.1, we project $\boldsymbol{h}$ with a linear layer $W_{\boldsymbol{D}} \in \mathbb{R}^{M \times |\mathcal{D}|}$ to obtain a distribution over $\boldsymbol{D}$. We then use a separate linear layer, $W_{\boldsymbol{U}} \in \mathbb{R}^{M \times |\mathcal{U}|}$, to project $\boldsymbol{h}$ and obtain a distribution over $\boldsymbol{U}$. To predict $\bar{\boldsymbol{Y}}$, we project $\boldsymbol{h}$ with a linear layer $W_{\boldsymbol{Y}}$. In the case of *GeMM*, we let $W_{\boldsymbol{Y}} \in \mathbb{R}^{M \times |\mathcal{D}|}$ in order to parameterize a mean of a *Log-Laplace* distribution for each dimension in $\boldsymbol{D}$. For *GeMM* U→Y, we set $W_{\boldsymbol{Y}} \in \mathbb{R}^{M \times |\mathcal{U}|}$ to output the mean of a *Log-Laplace* distribution for each unit in $\boldsymbol{U}$ and the remaining models, we set $W_{\boldsymbol{Y}} \in \mathbb{R}^{M \times 1}$ resulting in a single mean of a *Log-Laplace* distribution. For training, we use cross-entropy loss for the dimension and unit distributions, and the loss from the equation above for number prediction.

3.3 Dataset

We train and evaluate our models on *WiCo* [115], a dataset of English Wikipedia sentences where the number and unit in each sentence are human-annotated. We canonicalize the units and map each to a single dimension. For example both *feet per second* and *miles per hour* map to *velocity*. We show the distribution of all measurements and *lengths* in Figure 3.3. The resulting dataset consists of 919,237 sentences with annotated (number, unit, dimension) triples. We provide more details on the data in Appendix 3.8.1.

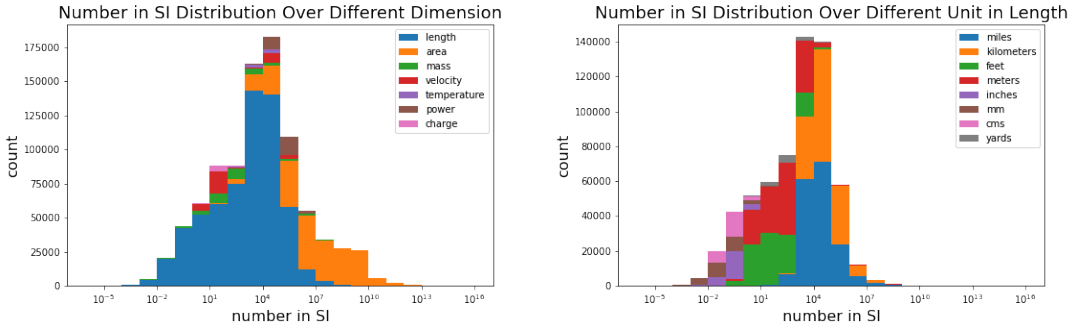


Figure 3.3: Histograms of *WiCo* numbers binned by base-10 exponent. All numbers are canonicalized to their SI form. **Left:** All numbers labeled by dimension. **Right:** Numbers in the *length* dimension labeled by unit.

Model	**10-shot**	**40-shot**	**70-shot**	**100-shot**
GeMM -Y -U ❄	15.5	50.0	52.5	53.4
GeMM -Y -U	**42.5**	**51.2**	**57.6**	**60.5**
Majority	14.3	14.3	14.3	14.3

Table 3.2: Results (measured by F1 ↑) of our few-shot experiment on dimension classification (probing $p(D|S)$). x-shot implies the model is trained on x labeled examples per dimension. **"mDiscD"** indicates an ablation of *GeMM* where $\boldsymbol{Y}$ and $\boldsymbol{U}$ are not modeled. ❄ indicates the model's parameters are frozen during training.

3.4 Experiments

We train all models using a batch size of 200 and use the AdamW [73] optimizer with a learning rate of $1e^{-4}$ and a linear warm-up schedule of 500 steps. We use the "❄" symbol to indicate that we freeze the transformer parameters for training. For all frozen models we use a log frequency weighted cross-entropy due to the highly imbalanced classes as well as a higher learning rate of $1e^{-3}$. We employ early stopping with a patience of five epochs on validation score.

To evaluate the performance of our models, we report the macro averaged F1 score for dimension and unit prediction and *log-mae* to evaluate number prediction. We define *log-mae* in Equation 3.2 where $\boldsymbol{Y}$ is the predicted number and $\boldsymbol{Y}^*$ is the ground truth number. As a simple baseline for dimension and unit prediction, we employ majority class voting. For number prediction we use the median of all the numbers in the training set.

$$\textit{log-mae} = \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{\mathcal{D}_{\text{test}}} |\log_{10} \boldsymbol{Y}^* - \log_{10} \boldsymbol{Y}| \tag{3.2}$$

Model	10-shot	40-shot	70-shot	100-shot
GeMM -U -D❄	1.94	1.82	1.72	1.75
GeMM -U -D	**1.70**	**1.56**	**1.43**	**1.41**
Median	1.99	1.99	1.99	1.99

Table 3.3: Results (*log-mae* ↓) of our few-shot experiment on number prediction (probing $p(Y|S)$).

Model	Probing Type	Val	Test
Majority	-	33.1	33.1
GeMM❄	$p(\boldsymbol{D}\|\boldsymbol{S})$	69.1	67.5
GeMM -Y -U	$p(\boldsymbol{D}\|\boldsymbol{S})$	88.0	86.8
GeMM -Y	$p(\boldsymbol{D}\|\boldsymbol{S})$	87.0	**87.3**
GeMM -U	$p(\boldsymbol{D}\|\boldsymbol{S})$	87.2	86.6
Lat-Dim	$p(\boldsymbol{D}\|\boldsymbol{S})$	9.0	9.1
GeMM	$p(\boldsymbol{D}\|\boldsymbol{S})$	87.4	87.0
GeMM U→Y	$p(\boldsymbol{D}\|\boldsymbol{S})$	86.4	86.1

Table 3.4: Results (**F1** ↑) for dimension prediction conditioned on $\boldsymbol{S}$ only. *GeMM* U→Y indicates a variant of *GeMM* where $\bar{\boldsymbol{Y}}$ is dependent on $\boldsymbol{U}$ (in addition to $\boldsymbol{S}$).

3.4.1 Few-Shot

To study the degree to which current pretrained models capture different aspects of numeracy, we consider the following few-shot experiment. We sample a balanced dataset of dimensions where each class gets 10, 40, 70, or 100 labeled examples. We train *GeMM* -Y -U and *GeMM* -U -D on the few-shot task where the pretrained text encoder $\boldsymbol{T}$ parameters are frozen and compare their performance against full fine-tuning. Due to the high variance of *GeMM* -Y -U, we report the average of three random seeds. In Table 3.2 and Table 3.3 we show results of *GeMM* -Y -U and *GeMM* -U -D respectively.

Although performance improves with more data, the frozen models significantly underperform their unfrozen counterparts across all dataset sizes. For example, in the 100-shot dataset, the frozen model shows 7.1 lower F1 and 0.34 higher *log-mae*. These results suggest that current pretrained transformers do not capture numeracy to a large extent.

3.4.2 Dimension Prediction

We train our models and their ablations on the full dataset and measure their performance on dimension prediction. In Table 3.4, we show the results of dimension prediction conditioned on $\boldsymbol{S}$. We observe that the performance gap between the frozen and unfrozen *GeMM* grows to 19.5 F1 on the test split despite training on 3 orders of magnitude more training data than the few-shot

Model	Probing Type	Val	Test
GeMM -U	$p(\boldsymbol{D}\|\bar{\boldsymbol{Y}}, \boldsymbol{S})$	95.5	95.7
GeMM U→Y	$p(\boldsymbol{D}\|\bar{\boldsymbol{Y}}, \boldsymbol{S})$	96.4	**96.6**

Table 3.5: Results (**F1** ↑) for dimension prediction conditioned on $\bar{\boldsymbol{Y}}$ and $\boldsymbol{S}$.

Model	Probing Type	Val	Test
Majority	-	8.9	9.0
GeMM❄	$p(\boldsymbol{U}\|\boldsymbol{D}, \boldsymbol{S})$	29.8	29.8
GeMM -Y	$p(\boldsymbol{U}\|\boldsymbol{D}, \boldsymbol{S})$	52.9	51.7
GeMM	$p(\boldsymbol{U}\|\boldsymbol{D}, \boldsymbol{S})$	51.5	**54.9**
GeMM U→Y	$p(\boldsymbol{U}\|\boldsymbol{D}, \boldsymbol{S})$	49.3	47.8

Table 3.6: Results (**F1** ↑) on unit prediction conditioned on the true dimension and text. Ablations are above the double horizontal line.

setting.

By using Bayes' rule, we perform dimension prediction conditioned on both $\boldsymbol{S}$ and $\bar{\boldsymbol{Y}}$ and show our results in Table 3.5. We observe that both models show improved dimension prediction ability when supplied with the number with *GeMM* U→Y reaching 96.6 F1 score, an effective error rate reduction of 75%.

3.4.3 Unit Prediction

We show the unit prediction performance of our models in Table 3.6. The strongest performing model for unit prediction was *GeMM* with a F1 score of 54.9. Again, the frozen *GeMM*❄ produced a 25.1 lower F1 score than its unfrozen counterpart.

We note that even though the F1 scores on unit prediction are much lower than dimension prediction, they are still significantly better than the majority baseline. Although one can freely substitute a unit with one in the same dimensional class, we tend to be more systematic and choose units that allow for more straightforward human readability or reflect the actual instruments used for measurement. As a result, we gravitate towards regularities that models can learn to recognize. The converse of this is also interesting as it suggests that the expressed units imply more semantic meaning than what is captured in the standardized measurement.

3.4.4 Number Prediction

We show the number prediction performance of our models in Table 3.7. Consistent with our previous experiments, all models outperform *GeMM*❄. Furthermore, we observe that not

Model	Probing Type	Val	Test
Median	-	1.98	1.97
GeMM ❄	$p(\bar{\boldsymbol{Y}}\|\boldsymbol{S})$	1.377	1.370
GeMM -U -D	$p(\bar{\boldsymbol{Y}}\|\boldsymbol{S})$	0.529	0.531
GeMM -U	$p(\bar{\boldsymbol{Y}}\|\boldsymbol{D},\boldsymbol{S})$	0.468	0.469
	$p(\bar{\boldsymbol{Y}},\boldsymbol{D}\|\boldsymbol{S})$	0.517	0.518
Lat-Dim	$p(\bar{\boldsymbol{Y}},\boldsymbol{D}\|\boldsymbol{S})$	0.545	0.546
GeMM	$p(\bar{\boldsymbol{Y}}\|\boldsymbol{S})$	0.517	**0.515**
GeMM U→Y	$p(\bar{\boldsymbol{Y}}\|\boldsymbol{U},\boldsymbol{D},\boldsymbol{S})$	0.401	**0.401**
	$p(\bar{\boldsymbol{Y}},\boldsymbol{U},\boldsymbol{D}\|\boldsymbol{S})$	0.526	0.526

Table 3.7: Results (*log-mae* ↓) for number prediction conditioned on $\boldsymbol{S}$. In the second row of *GeMM* -U, we select the highest scoring $d^* \in \boldsymbol{D}$ and predict y conditioned on d^* and $\boldsymbol{S}$. In the second row of *GeMM* U→Y, we select the highest scoring $u^* \in \boldsymbol{U}$ and $d^* \in \boldsymbol{D}$ and predict y conditioned on u^*, d^*, and $\boldsymbol{S}$. For *Lat-Dim*, we sum over the latent variable $\boldsymbol{D}$ to predict y conditioned on $\boldsymbol{S}$.

modeling $\boldsymbol{U}$ and $\boldsymbol{D}$ (as is the case in *GeMM* -U -D) increases *log-mae*, i.e., results in worse numerical prediction. While competitive with *GeMM* and its variants on number prediction, *Lat-Dim* cannot predict dimensions with the same efficacy (Table 3.4).

We also experiment with the setting where *GeMM* -U conditionally generates the number for a particular dimension. In this setting, *GeMM* -U improves *log-mae* to 0.469. Extending this setting further, we condition *GeMM* U→Y on both a unit and a dimension to produce the best *log-mae* among our models: 0.401.

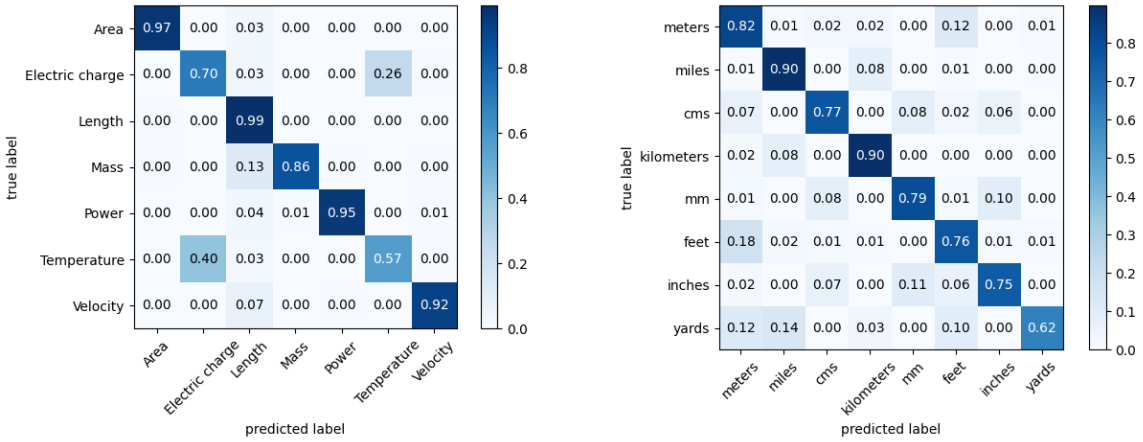
We now revisit our original motivating example: *"Alex Honnold climbed for [NUM] [UNIT]"*. Assume we want to know the distance of a climb. To do this, we condition *GeMM* U→Y on $\boldsymbol{D} = length$ and $\boldsymbol{U} = feet$. If, on the other hand, we want to know the duration of a climb, we change the conditioning to $\boldsymbol{D} = time$ and $\boldsymbol{U} = hours$. Now, if we want to know the length of Alex Honnold's climbing career, we condition *GeMM* U→Y on $\boldsymbol{D} = time$ and $\boldsymbol{U} = years$. These examples illustrate the flexibility of *GeMM* U→Y and the importance of jointly modeling numbers, units, and dimensions.

3.4.5 Quantitative Analysis

Dimensions and Unit

In Figure 3.4a we visualize a confusion matrix of dimension predictions by *GeMM* U→Y. The low accuracy for electric charge and temperature is attributed to a mislabeling in the dataset.[2] For mass, we find many ambiguous situations where either mass or length are appropriate. See the

[2]Sentences with mislabeled `Celsius` as `Coulombs`, which may due to wrong annotation between `°C` and `C`. Also observed by Elazar et al. [40]



(a) (b)

Figure 3.4: Confusion matrices for predictions by *GeMM* U→Y over the validation split. **Left 3.4a:** Dimension prediction. Most misclassified dimensions are similar to their ground truth counterparts in terms of Manhattan distance. **Right 3.4b:** Unit prediction for examples that share the *length* dimension. Most misclassified units of length share similar magnitudes to their ground truth units.

first row of Table 3.10 for such an example.

Thus far, we have treated dimensions as distinct classes with no relationships. However, dimensions are compositions of the seven fundamental dimensions. Therefore, dimensions that share fundamental dimensions are more similar than those that do not. To quantify this similarity, we can treat dimensions as a vector where each element represents the exponent of a fundamental dimension. Then to measure the similarity of two dimensions, we take their Manhattan distance. To illustrate, assume there exist only two fundamental dimensions: Length and Time. Let $speed = (1, -1)$ and $length = (1, 0)$ where the first element represents Length and the second represents Time. The Manhattan distance between $speed$ and $length$ is equal to one. In Figure 3.5, we visualize the Manhattan distance between the predictions of *GeMM* U→Y and ground truth. We observe that there is generally an inverse relationship between error count and the distance of the errors. This observation suggests that our model has learned that some dimensions are more similar than others. This suggestion is reinforced by Figure 3.4a where misclassifications tend to have small distances from the true dimension. For example, velocity is most often misclassified as length. For unit prediction, we find that most mistakes occur substituting units with ones that have similar magnitudes like feet for meters or kilometers for miles.

Numeracy

In Table 3.8, we show *log-mae* by dimension as predicted by *GeMM* U→Y. We note that errors are not uniform across dimensions, predicting *areas* is 2.2 times harder *velocities*. We also observe that the magnitudes of errors seem to be positively correlated with the variances observed in Figure 3.3.

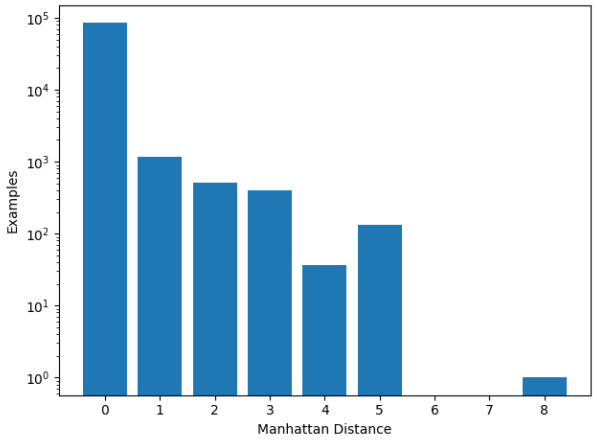


Figure 3.5: Manhattan distance between true and predicted dimensions by *GeMM* U→Y. We treat dimensions as vectors whose elements are the exponents of the fundamental dimensions that compose a given dimension. Note that the y-axis is in log-scale.

Length	Area	Velocity	Mass	Power
0.37	0.54	0.19	0.55	0.27

Table 3.8: *log-mae* ↓ by dimension. It is harder to predict numbers of Area and Mass than other dimensions.

Human Evaluation

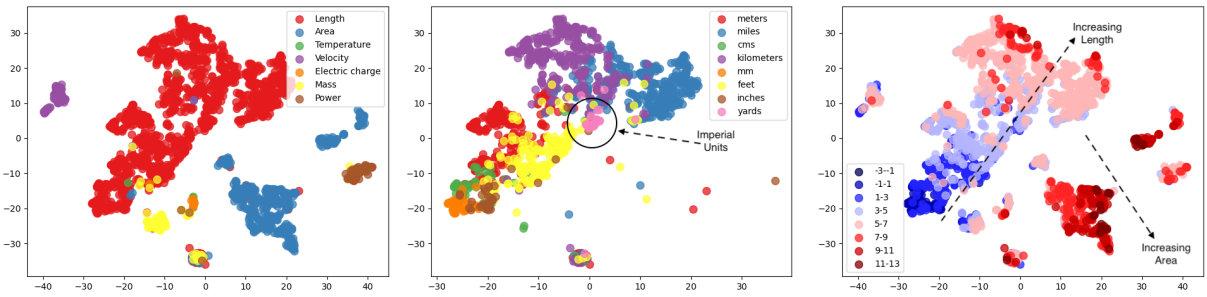
We perform two evaluations of *GeMM* U→Y against human annotators. In the first evaluation, we compare against the combined effort of three Technical Annotators on a balanced set of 90 sentences randomly sampled from the test set. The annotators worked together to predict the missing dimensions, units, and accurate measurement estimates. Examples of sentences and annotations shown in Table 3.10.

In the second evaluation, we compare against Amazon Mechanical Turk (AMT) Annotators on a balanced set of 2,122 sentences randomly sampled from the test set. We show the results for both evaluations in Table 3.9.

In both evaluations, the model outperforms the human annotators on every task. For dimension prediction, the model led by 7.4-7.8 percentage points. Of the sentences where the dimension was correctly annotated, the model led by 33.5-39.9 percentage points on unit prediction. For sentences where both the model and human correctly predicted the dimension, the model predicted a number closer to ground truth 66.2-78.8% of the time.

	Model		Human		Model > Human
	D	***U***	***D***	***U***	***Y***
Tech Ann.	**96.7**	**86.2**	88.9	46.3	**78.8**
AMT Ann.	**96.7**	**77.0**	89.3	43.5	**66.2**

Table 3.9: Dimension and unit prediction accuracy of our human evaluation experiment. *GeMM* U→Y outperformed the human annotators in both evaluations. **Tech Ann.** is over a balanced set of 90 sentences labeled by Technical Annotators. **AMT Ann.** is over a balanced set of 2,122 sentences annotated by AMT Annotators. The final column shows the model predicted a number closer to ground truth in 66.2-78.8% of the cases.



(a) (b) (c)

Figure 3.6: t-SNE visualizations of semantic head embeddings labeled by **(left 3.6a)** dimension, **(middle 3.6b)** units of *length*, and **(right 3.6c)** number exponent bin. **Middle**: we observe a clustering of imperial units: feet, yards, miles. **Right**: we show two directions where magnitudes of length and area measurements increase in value.

3.4.6 Qualitative Analysis

Semantic Head Embeddings

In Figure 3.6 we plot the t-SNE embeddings of the sentences' $\boldsymbol{h}$, the output of our text encoder. We label each $\boldsymbol{h}$ with the masked measurement's true dimension, unit and exponent of the number. In 3.6a we observe that most embeddings labeled by their true dimension tend to form tight clusters. In 3.6b we filter to only show embeddings that share the *Length* dimension and label them by their units. We find that clusters are organized by the relative magnitudes of their units: large (*Kilometers, miles*), medium (*feet, meters*), and small (*millimeters, inches, centimeters*). Further we see that *yards* appear close to other *imperial units* of *feet* and *miles*. Finally, in 3.6c when embeddings are binned by the exponent of their values we observe that the left to right direction appears to capture the increasing magnitude of a number.

#	Text	True			*GeMM* U→Y Prediction			Human Prediction		
		Dim	Unit	Num	Dim	Unit	Num	Dim	Unit	Num
1	Hope is gaff rigged, 'V'-bottomed and has an [#NUM] [UNIT] centerboard.	Mass	pounds	385.6	Length	feet	2.97	Length	meter	50
2	Some have been running for over 50 years, each covering about [#NUM] [UNIT].	Velocity	$\frac{\text{miles}}{\text{year}}$	0.10	Area	sqkm	2.09E+10	Area	sqmi	2.59E+07
3	Another medium-sized corvid, the [#NUM] [UNIT] Eurasian magpie (Pica pica) is also amongst the most widely reported secondary prey species for goshawks there.	Mass	grams	0.22	**Mass**	**grams**	0.05	**Mass**	**grams**	**0.2**
4	The twin cylinder, liquid-cooled, in-line two-stroke, [#NUM] [UNIT] Rotax 582 has also been used.	Power	horse-power	47725	**Power**	**horse-power**	39248	**Power**	**horse-power**	**45000**
5	Chrysothamnus may grow up to a [#NUM] [UNIT] tall shrub or subshrub, usually with woody stem bases	Length	cms	1.2	**Length**	meters	**1.147**	**Length**	meters	1
6	Kurt Busch was the fastest in the first practice session with a time of 21.372 seconds and a speed of [#NUM] [UNIT].	Velocity	$\frac{\text{miles}}{\text{hour}}$	75.1	**Velocity**	$\frac{\textbf{miles}}{\textbf{hour}}$	**63.584**	**Velocity**	$\frac{\text{meters}}{\text{second}}$	10

Table 3.10: Instances of the *MMP* task performed during our human evaluation experiment, all numbers are in SI units. In ex. 1, both the model and humans predict the incorrect dimension length instead of mass. The preceding sentence of ex. 2 references "trains" leading both to incorrectly predict area instead of velocity. In ex. 6 the model predicts the speed of the NASCAR driver Kurt Busch's car whereas the humans had mistaken him for a runner.

3.5 Related Work

3.5.1 Numeracy

Multiple works have probed word embeddings like word2vec, GloVe, FastText [84] and contextual embeddings from models like BERT [130, 145] or T5 [86] on a variety of numerical tasks like sorting, numeration, magnitude prediction, and common sense [66]. Several works have targeted numeracy pretraining using left to right language models [107], CNN and RNN based models [28], pretrained transformers [51, 108], for an overview [116].

Incorporating synthetic mathematical data augmentations [43] has improved question answering while numerical pretraining has been shown to lower masked language modelling perplexity [115]. Either directly or indirectly units have been involved in providing more interpretable explanation of quantities [26], solving Fermi problems [53] and resolving numeric Fused-Heads [39].

Numeracy Benchmarks

Several numeracy benchmarks have been proposed like quantitative reasoning in natural language entailment [90] and synthetic measurement estimation [51]. The closest benchmark to our work is the Distribution over Quantities dataset (DoQ) introduced by Elazar et al. [40]. A rule-based method was combined with simple heuristics to build DoQ resulting in its high-coverage albeit also higher noise. Although, *WiCo* is smaller, it has much higher fidelity since it utilizes a feature used by editors of Wikipedia to automatically convert quantities into different units. Further, *WiCo* provides the whole sentence as context as opposed to triplets of words. Zhang et al. [145] use artificial templates to probe models on DoQ and find little difference between numerically

pretrained and frozen embeddings such as ELMo. In contrast, our findings show there is a significant gap on *WiCo* between fully finetuned models and their frozen counterparts.

3.6 Limitations

The pretrained RoBERTa model we use in experiments was likely pretrained on data that included *WiCo*. Thus, it is reasonable to be concerned about inflated test performance. That said, the task we consider is distinct from the self-supervised task used to pretrain RoBERTa (i.e. masked word classification vs. masked number regression). Further, our experiments on directly probing RoBERTa to predict masked numbers and units showed poor performance – indicating, perhaps, that even if RoBERTa's pre-training set did include *WiCo*, RoBERTa did not memorize aspects of our test set relevant to masked number prediction, partially mitigating these concerns.

The human evaluation studies we conducted are a quite limited 'guesstimating' task. The human annotators were not allowed to use any external information from searching the internet or looking up answers in knowledge-bases. Their total average completion time per question was 33 seconds. Furthermore, many annotators may not have strong intuition about measurements with unfamiliar and uncommon unit types. For these reasons it is not surprising that our models outperform the human annotators in this limited experiment. However, these human evaluation studies do help calibrate the difficulty of the *MMP* task on *WiCo*.

3.7 Conclusion

In this work we propose Masked Measurement Prediction, a new task that requires models to jointly predict masked numbers and units in running text. We motivate this task as an important extension of existing masked number-only prediction tasks that addresses their limitations and allows for better evaluation of numeracy in NLP models. In our study, we show that probing of traditional pretrained transformers exposes a gap in their understanding of contextualized quantities. Through careful quantitative and qualitative analysis of our new model, which directly reasons about underlying units and dimensions, we find that it is possible to learn good representations of measurements. For future work we aim to extend this dataset to cover more existing standardized units from organizations such as UNECE.[3] We hope our *MMP* task encourages research into further development of better numeracy methodologies.

3.8 Appendix

3.8.1 Dataset

We train and evaluate our models on Wiki-Convert (*WiCo*) [115], a dataset of English Wikipedia sentences where the number and unit in each sentence are human-annotated. The built-in template in Wikipedia can ensure the text contains numbers and units. For example, `{{convert|2|km|mi}}`

[3]United Nations Economic Commission for Europe

Input: [UNIT]	m	km	ft	mi	yd	in	meters	kilometers	feet	miles	yards	inches	-
Output	200	10	200	2	100	1	200	20	20	2	50	3	-
Convertion factor	1	1000	0.3048	1609.34	0.9144	0.0254	1	1000	0.3048	1609.34	0.9144	0.0254	-
Metric Output	200.0	10000.0	60.96	3218.68	91.44	0.0254	200.0	20000.0	6.096	3218.68	45.72	0.0762	-
Mean (Metric Output)							-						3086.8 m
std (Metric Output)							-						5820 m

Table 3.11: Example outputs for **Alex Honnold climbed for [MASK] [UNIT]**.

displays as `2 kilometres (1.2 mi)`. By searching within Wikipedia articles for the use of this template, the authors of *WiCo* automatically extract human-annotated numbers. To perform unit canonicalization, we use Pint [4] whenever the mapping is unambiguous. In the ambiguous case, we manually inspect the sentence and perform the mapping. For example, we map the unit `sqmi` in *WiCo* to `square miles` to let pint perform unit canonicalization. Table 3.10 shows examples of the extended dataset. The original dataset contains 924,473 sentence. The median sentence length is 106 characters, with 29,597 sentences has a length shorter than 20 characters. We provide statistics of the data in Table 3.1. For preprocessing we exclude sentences which have more than 64 tokens to have efficient computing memory or where the number is negative for simplicity. According to Thawani et al. [115] *WiCo*, "... has been extracted from Wikipedia dumps, which are licensed under the GNU Free Documentation License (GFDL) and the Creative Commons Attribution-Share-Alike 3.0 License." Thawani et al. [115] constructed *WiCo* with the intent that it be used to further numeracy NLP research. Our use of *WiCo* is aligned with its authors' goals.

3.8.2 MLM Preliminary Unit Probe

We perform a preliminary unit probe shown in Table 3.11. The model predicts vastly different numbers when conditioned on different units. We observe a mean of 3086.8 and a standard deviation of 5820 for all the converted metric output.

3.8.3 Experiments

We train our model *GeMM* U→Y on a single Nvidia GeForce RTX 2080 Ti for 4 hours and 14 minutes with a total parameter of 124,696,538.

Quantitative Analysis

In Figure 3.7, we show *log-mae* is relatively small for small magnitude units, which means predicting numbers for small magnitude units is easier than predicting numbers for their larger counterparts.

In Figure 3.4, we show confusion matrices of dimension and unit predictions by *GeMM* U→Y.

[4]Pint: https://github.com/hgrecco/pint

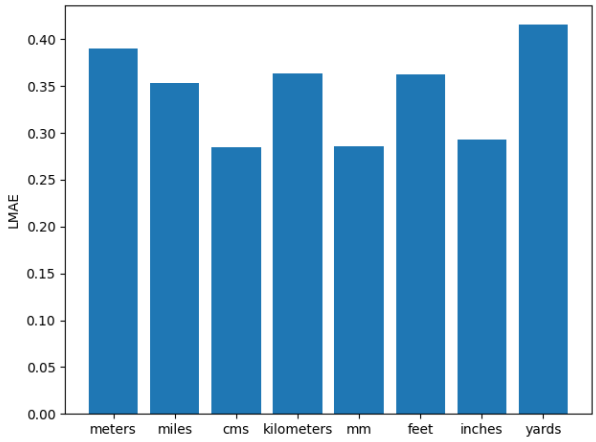


Figure 3.7: *log-mae* ↓ by units of length. Predicting numbers for small magnitude units is easier than predicting numbers for their larger counterparts.

3.8.4 Human Annotators

Evaluation 1

The Technical Annotators have diverse scientific backgrounds ranging from chemistry, earth sciences, and computer science. One annotator is a native Chinese speaker, and two are native English speakers.

Evaluation 2

In Figure 3.8 we show the instructions provided along with the interface we designed for our *MMP* task. While the workers' geographic location were not provided to us by Mechanical Turk, we aimed to compensate the workers above the US federal minimum wage of $7.25. We paid workers $0.15 per annotation with an average completion time of 33 seconds. This equates to an hourly rate of $12.80 after Mechanical Turk fees. Other demographic information is only provided by Mechanical Turk for an extra fee.

3.8.5 Ethical Considerations

Like any system that makes predictions, those made by *GeMM* are not necessarily accurate and may be used by malicious actors to generate fake information to mislead their audience. Additionally, *GeMM* is an extension of RoBERTa and therefore inherits the biases learned during the training of RoBERTa. Our work focuses exclusively on English and Arabic numerals. As noted by Thawani et al. [115], the units in *WiCo* are heavily biased towards European and American units as they are over-represented in English Wikipedia.

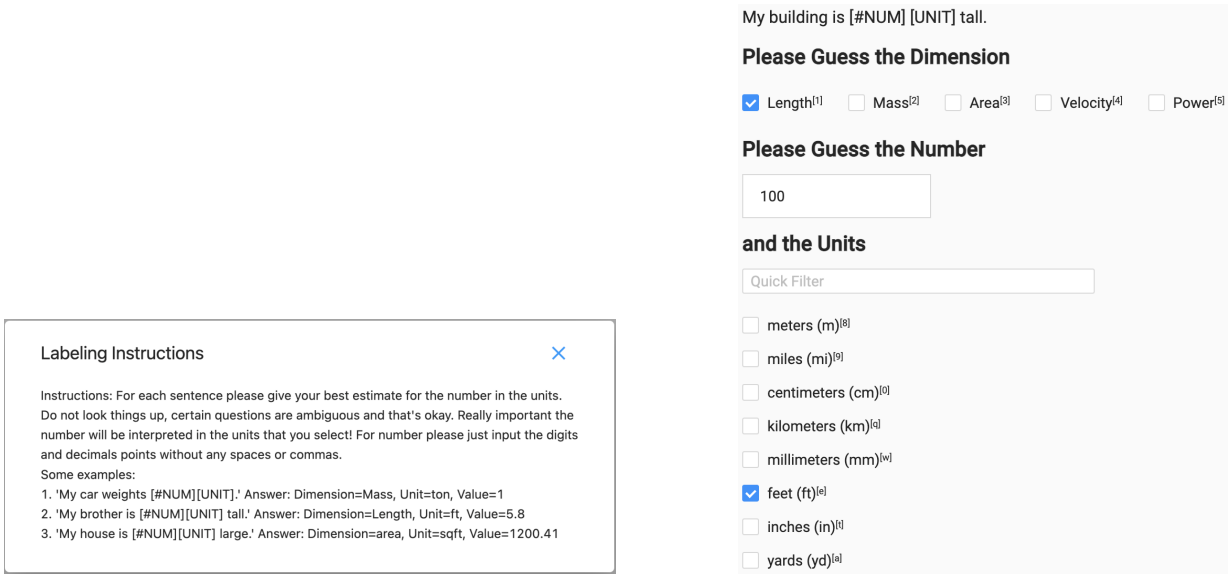


Figure 3.8: **Left**: Instructions for labeling task. **Right**: we show the interface used by the labelers

Chapter 4

Numerical Correlation in Text

- Daniel Spokoyny and Chien-Sheng Wu and Caiming Xiong, “Numerical Correlation in Text.”, In Proceedings of EMNLP Workshop on Mathematical Natural Language Processing 2022

4.1 Introduction

Numerical reasoning tasks are one area where the performance of Large Language Models (LLMs) has not improved as drastically [88] as on other tasks. Good performance is critical for many downstream applications in areas such as fact checking, question-answering, or search. Different tasks have been proposed to evaluate the numerical reasoning capabilities of LLMs [82].

We can analyze these tasks along two dimensions: diversity of knowledge required and how solvable the task is. Higher diversity ensures better coverage across different domains while higher solvability yields more interpretable metrics. Mathematical word problems (MWPs) are written in a way that the text of the problem is always sufficient to determine the exact unique answer and are therefore highly solvable. However, they lack in diversity since many MWP datasets are constructed from templates or are even fully synthetic.

In contrast numerical cloze-style problems requires highly diverse knowledge since they can be easily formed from any text that includes numbers. A consequence of formulating cloze-style problems is that many texts do not provide sufficient information to determine the correct answer and have inherent uncertainty which results in a lower solvability. As an example from the NumerSense dataset [66], "Some plant varieties can grow up to <mask> feet tall." In Figure 4.1 we show an illustrative plot of tasks along these two dimensions. A good numeracy evaluation task should be both diverse and solvable.

In this work we propose Numerical Correlation, a new task that aims to retain both high diversity and high solvability. Given two numbers in text the task is to predict whether the numbers are positively, negatively or not correlated. For example: “Some plant varieties can grow up to 6 feet tall and require 20 liters of water a month”. We expect a positive correlation between the height of the plant and the amount of water it would need. This shows the key insight that predicting the correlation relationship between two numbers is possible without having to exactly predict the missing numbers. The task of numerical correlation requires a variety of commonsense

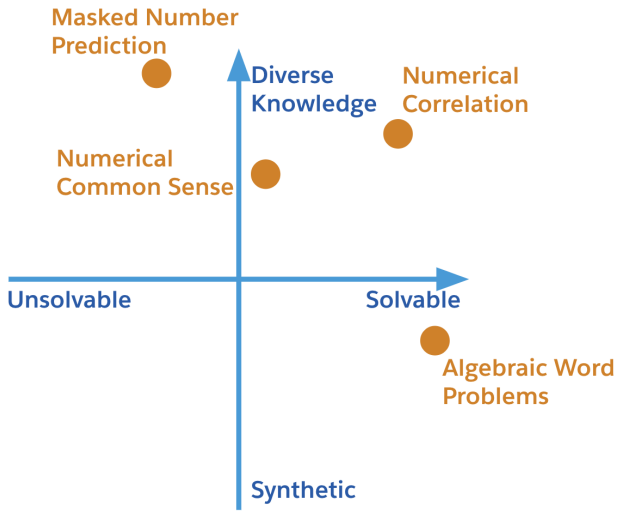


Figure 4.1: An illustrative plot of certain numerical evaluation tasks along the two dimensions of diversity and solvability. Our aim with numerical correlation is for the task to be both diverse and solvable.

# Ex	Text	Label
1.	The president travels on average **30** times a year on Air Force one a Boeing **747**.	Neutral
2.	A **2** bedroom, **1800** square feet house is hard to find in this neighborhood.	Positive
3.	To cook a 20 lb turkey place in the oven for **2** hours at **435** degrees.	Negative

Table 4.1: Explanations for the three examples: 1) the model of the plane should not change how often the president travels, 2) we expect more bedrooms to increase the size of the house, and 3) we expect an increase of temperature to decrease the cooking time.

reasoning skills but is trained with a cross-entropy objective and evaluated with a simple accuracy metric. We provide examples of sentences and their labels in Table 4.1.

Although correlation between two numbers can involve incredibly complex functions, we approximate the correlation to be linear and treat it as a three-way classification. We use a qualification task to select a group of Amazon Mechanical Turk (AMT) labelers and construct a dataset of Wikipedia sentences which contain two numbers and their correlation relationship.

We investigate the performance of four models: two general pretrained language transformers and two numerically aware models on our new dataset in a few-shot setting. When probed on the numerical correlation task we see that all models exhibit a plateau in their performance with only 6% of the training data. Further all models underperform the human baseline in both the finetuning and linear probing setting. Surprisingly, our results also indicate that existing numerically pretraining methods do not result in better performance on the numerical correlation task.

	Test F1 (Neutral, Positive, Negative)				
w/ 10% Train	GenBERT	GeMM	RoBERTa-Base	Bart-Large	Human Jackknife
Linear Probing	33.0 (71.1 / 26.7 / 0.1)	37.9 (72.3 / 41.6 / 0)	23.7 (71.1 / 0 / 0)	**64.9** **(76.6 / 57.7 / 60.4)**	~77
Finetuning	62.1 (77.5 / 59.3 / 49.6)	66.7 (77.9 / 65.5 / 56.8)	**69.6** **(80.7 / 66.3 / 61.8)**	68.6 (* / * / *)	

Figure 4.2: Summary of the performance of the four models on the numerical correlation task with 10% of the training data.

4.2 Dataset

4.2.1 Qualification

We used ten handwritten numerical correlation examples and had 100 AMT workers with 99%> approval rate label them. On average each question took around 1 minute to complete. Thresholding on 80% accuracy or above left us with 18 AMT-labelers. Examples are shown in Table 6.1 and instructions given in Figure 4.5.

4.2.2 Annotation

We use the WikiConvert dataset [115] which contains over 900k sentences with at least one measurement in each sentence. We use the three original correlation labels (Positive, Negative, Neutral)[1] and had each sentence labeled by three different AMT-labelers. We selected 1,000 random sentences that contain two measurements and another 1,000 sentences that contain any two quantities.[2]

We used Krippendorff alpha to measure the inter-annotator agreement and found that the agreement was 0.55 (scale is [-1,1]). We computed an average "Jackknife" F1 score of 77 by choosing one label to be the ground truth and averaging the F1 score of the other two labels. We also observe that the time taken to label each sentence rose to 1.7 minutes on average, likely due to the increased difficulty to ascertain the correlation in random sentences.

Negative

Out of the 2,000 sentences only 42 were found to have a negative correlation which is too few data points to train or evaluate a model. For this reason we experimented with two strategies to generate more negative correlation examples: 1) editing a measurement in real sentence 2) providing a description of a real negative relationship and prompting labelers to provide a sentence as an

[1]We introduce a fourth label (Unanswerable) which we advised the labelers to use sparingly when they were unsure of the answer

[2]We filtered out sentences that contained dates or where shorter than 64 characters in length.

example. In a small pilot we found that the first strategy was incredibly more time consuming to complete and so we only used the second strategy to generate negative correlation examples. We provided 60 descriptions of negative relationships and asked the three labelers to provide an example for each sentence.[3] In total our dataset consists of 124 sentences with negative correlation, 746 with positive correlation and 1,155 with neutral correlation.

4.3 Experiments

Given a sentence X and two numbers y_1 and y_2 in the text, we define the task of predicting the correlation between the two numbers as a classification task with the label set $C = \{Positive, Negative, Neutral\}$. We compare four models, two general pretrained language models (BART [65] and RoBERTa [70]) and two numerically aware models (GeMM [109] and GenBERT [44]). We conduct few-shot learning experiments where the model is trained on between 1% to 10% of the training data and the remaining data is split into a validation and test set evenly. We train all models with the AdamW optimizer [73] with a learning rate of 1e-5 and a batch size of 16. We use the majority vote labeling to choose the final label for each sentence in all subsequent experiments.[4] We report the test F1 scores averaged over 5 initialization seeds.

4.3.1 Supervised

We conduct few-shot linear probing as well as full finetuning experiments and plot the results in Figure 4.3 and Figure 4.4 respectively. For our linear probing experiments we freeze the parameters of the model and only train a linear classifier, $W_\theta \in \mathbb{R}^{d\times 3}$, where d is the hidden size of the model. We observed that BART performed better by a large margin (20 F1) as compared to the second best performing model, GeMM. However, all models experience a plateau in performance after only 6% of the training data.

Unlike the linear probing experiments, when we finetune the models we observe that all models (except GenBERT) converge to similar performance, approximately 10 F1 points below human performance. The poor performance of GenBERT could be explained by the fact that it uses a BERT architecture whilst the other models are based on RoBERTa and BART. We present all of the supervised Test-F1 results with 10% of the training data in Figure 4.2.

4.3.2 Unsupervised

Since we observe the actual values of the both numbers we can probe a model in an unsupervised fashion to predict the correlation relationship. We do this by selecting one number (y_1) to be the target prediction and masking its value in the sentence. We then probe the model to predict the value of the target (y_1) with different values of the other number (y_2). We use GeMM, a numerically pretrained model [109] and denominate the model's prediction for the masked value as $\hat{Y}$.

[3] We hand filtered out sentences that did not properly follow the instructions.

[4] In case of a tie we do not use the sentence in our data.

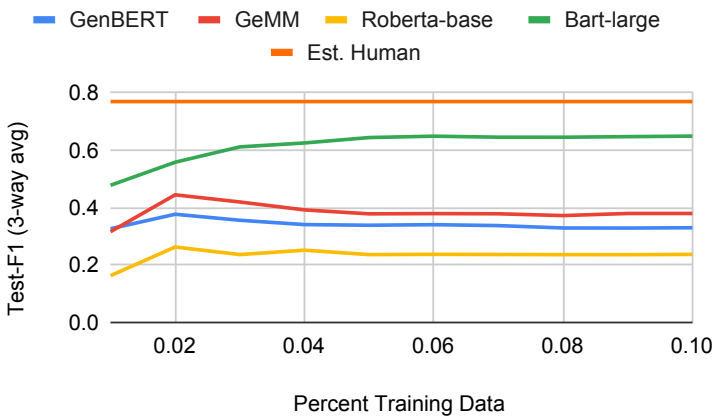


Figure 4.3: Linear probing experiments with 1% to 10% of the training data.

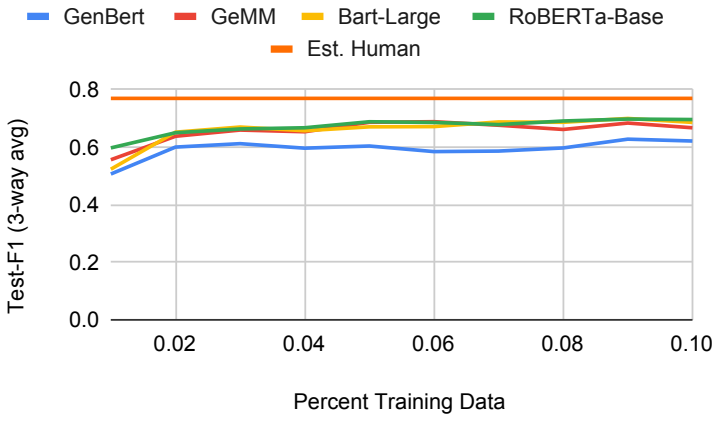


Figure 4.4: Full finetuning experiments with 1% to 10% of the training data.

We construct $\mathcal{N}$ examples, $\{X_1, X_{\mathcal{N}}\}$, by selecting values linearly spaced between $\{y_2 * 0.5, y_2 * 2\}$ and pass each example to the model to predict the $\mathcal{N}$ values of $\{\hat{Y}_1, \hat{Y}_{\mathcal{N}}\}$. We can then calculate the R-squared values of the linear regression for each pair of numbers in a sentence. We pick a threshold value τ and build a deterministic classifier which predicts "Neutral" if the R-squared value is less than τ, "Positive" if the R-squared value is greater than τ and the slope is positive, and "Negative" if the R-squared value is greater than τ and the slope is negative. When evaluated on a held out test set this classifier performs close to randomly guessing the label.

4.4 Related Work

4.4.1 Numerical Reasoning

An active area of research in NLP is focused on solving numerical reasoning tasks. There have been many datasets collected such as AQuA-RAT [68], Dolphin18K [49], Math23K [134], MathQA [4] which contain a mathematical question expressed in natural language and an answer. Benchmarks which aim to evaluate the general abilities of LLMs like BIG-bench, have also incorporated numerical reasoning tasks such as arithmetic questions or unit conversion [110]. To solve these problems a model needs to perform certain necessary calculations to arrive at the answer. Typically the value of the numbers provide no information to help disambiguate the derivation of the solution and can be treated symbolically. One key aspect of these tasks is that there exists no ambiguity in the answer.

4.4.2 Commonsense Reasoning

Another area of research has focused on cloze-style prediction of numbers in textual contexts. Certain works have limited the output space of numbers to small ranges [66], their exponent value [28] whilst others have aimed to produce distributions over the entire real number line [106, 108]. As opposed to the previous section, these tasks commonly do not have a correct answer but are ambiguous. A great advantage of numerical cloze-style reasoning is the ubiquity of available data in different forms and domains. However, it is difficult to measure progress and interpret the evaluation metrics such as likelihood for these types of commonsense tasks.

There are other NLP tasks which have concentrated on the difficulties that arise when numbers are present in a text. Ravichander et al. [91] proposed EQUATE, a benchmark quantitative reasoning in natural language inference while other works have focused on quantity entailment [98]. Dubey et al. [37] built a dataset where the numerical values were useful to predict the sentiment of sarcastic tweets. Sundararaman et al. [112] proposed a classification task of numbers into entities (Count, Size, Year, Percentage, Date, Age), while similar work has considered the problem of solving numeric Fused-Heads [38]. Our work on the correlation task focuses on a particular relationship between two quantities in text. However there are others potential relationships between numbers in text that could be explored such as causation.

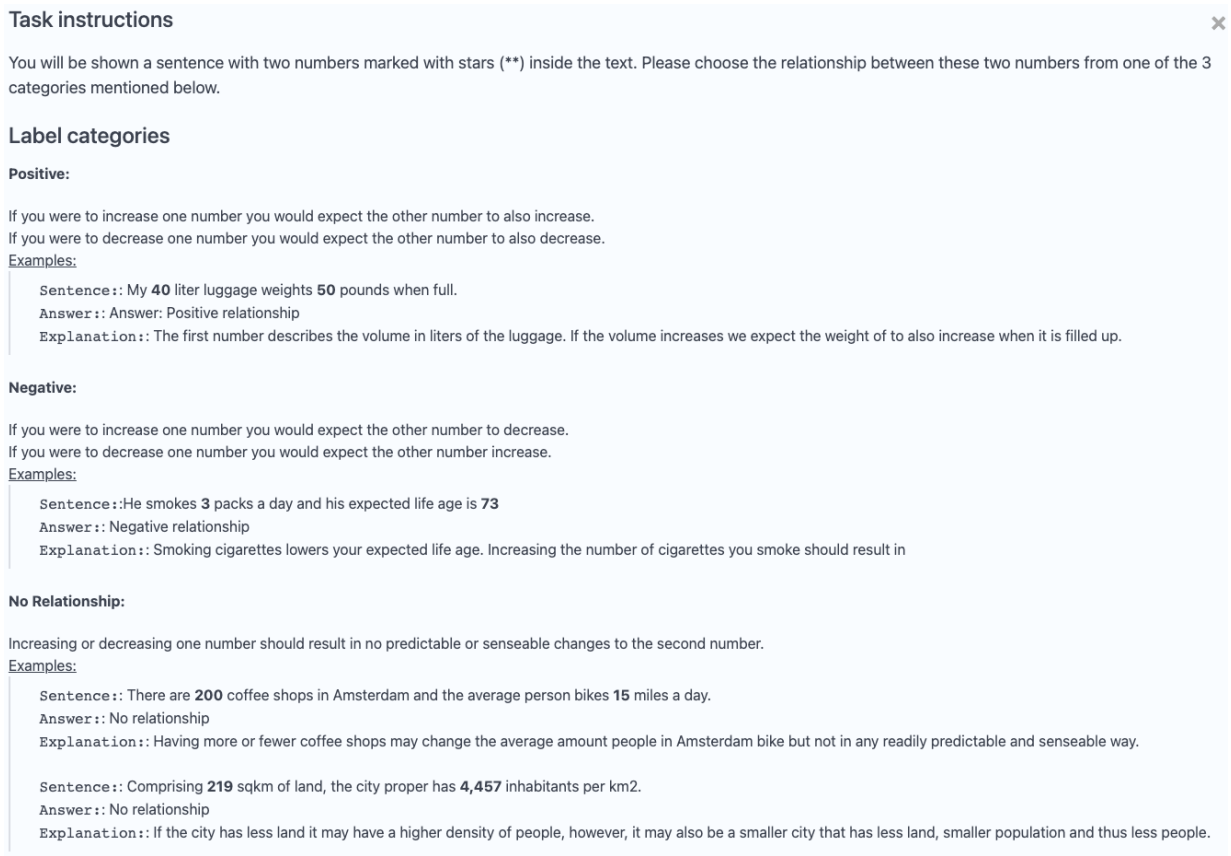


Figure 4.5: Instructions given to the labellers for the qualification task.

4.5 Conclusion

We introduced a new task of predicting numerical correlation in text and build an annotated dataset to evaluate models on this task. Using this dataset we show that pretrained language models have poor performance on this task and that current methods to add numerically aware pretraining to models are not effective. We identified that there exists a large gap between human performance and the best supervised model. In the future we hope to expand our annotation to include the slope of the correlation. We believe that predicting both the slope and correlation type of two numbers can be improve interpretability in numerical question answering and commonsense reasoning applications. In future work we also plan to expand the dataset to capture numerical correlation relationships in longer chunks of text such as paragraphs and documents.

4.6 Appendix

# Ex	Text	Label
1.	I wear my nike shoes out in only **3** months because the soles are only **1/2** an inch thick.	Positive
2.	To cook a 20 lb turkey place in the oven for **2** hours at **435** degrees.	Negative
3.	Jordan trained for his race by running **5** miles at a pace of **10** mph.	Negative
4.	The president travels on average **thirty** times a year on Air Force one a Boeing **747**.	No Relationship
5.	My house has **2** bedrooms and is **1800** square feet.	Positive
6.	Blackthorn was one of **39** original **180** feet seagoing buoy tenders built between 1942-1944.	No Relationship
7.	The family bought a **two** ton pickup truck with 180 hp and a fuel efficiency of **25** miles per gallon.	Negative
8.	My subaru has a **4** cylinder and **150** horse power enginer.	Positive
9.	Like all Type UB III submarines UB-102 carried **10** torpedoes and was armed with a **10** cms deck gun.	No Relationship
10.	The Triple Crown of Canoe Racing consists of three separate marathon races with a total distance of **308** miles over **5** days of racing.	Positive

Table 4.2: The ten examples used to qualify AMTworkers.

Chapter 5

BERT Classification of Paris Agreement Climate Action Plans.

- Tom Corringham, Daniel Spokoyny, Eric Xiao, Christopher Cha, Colin Lemarchand, Mandeep Syal, Ethan Olson, Alexander, "BERT Classification of Paris Agreement Climate Action Plans.", In Proceedings of ICML 2021 Workshop on Tackling Climate Change with Machine Learning.

5.1 Introduction

The United Nations Framework Convention on Climate Change (UNFCCC) is a global framework for addressing the challenges of anthropogenic climate change [62]. Under the 2015 Paris Agreement each UNFCCC signatory agreed to submit a Nationally Determined Contribution (NDC) upon ratification of the agreement by the country's national government [122]. These climate action plans set objectives and timelines for reductions in greenhouse gas emissions for each country. The documents, while often aspirational in nature, provide useful information about the challenges facing each country, their stance towards climate change, and their ambitions regarding mitigation efforts [121]. Here, a deep learning model is applied to 165 of these documents to build a sentence classifier which could be used to generate policy-relevant metrics over a wide range of documents.

The volume of information contained in climate policy documents is growing rapidly. Every year large stakeholders such as governments and corporations produce text-based climate assessments and action plans to communicate and satisfy regulatory requirements. NLP and machine learning can be used to provide data for climate policy analysis, improve tools for evaluating policy, and provide new tools for policy assessment [95]. Supervised and unsupervised NLP content analysis methods have been used to analyze political texts [45] including climate negotiation texts [9, 99, 129], climate adaptation analyses [15], and corporate climate financial disclosures [76].

BERT [34] is a bidirectional transformer model that has been pretrained on a large corpus of textual data using the masked language modeling objective. BERT and other pretrained

transformer models through finetuning have achieved state-of-the-art results in a variety of NLP tasks [94] including sentence classification. This is makes them good candidates for climate change text applications where large labeled data sets are currently unavailable.

Recently, the BERT model has been used to extract information from climate-related regulatory disclosures. Varini et al. [127] use BERT to classify sentences from U.S. Securities and Exchange Commission (SEC) filings as climate related or not climate related. Kölbel et al. [56] apply BERT to SEC filings to distinguish between sentences that discuss physical climate risk (e.g., due to sea level rise or extreme weather events) and transition risk (due to expected changes in climate-related regulation). Using the classified text they develop metrics that they relate to credit default swap rates. Bingler et al. [17] use BERT to classify sentences and paragraphs from corporate risk disclosure documents into four categories to assess the impact of the Task Force on Climate-related Financial Disclosures (TCFD).

Here, BERT is applied in a similar way to classify sentences in national climate action plans. Metrics derived from BERT-classified climate documents could be used to investigate the relationships between document characteristics and country characteristics such as exposure to climate risks or energy and resource endowments. Understanding these relationships could be used to evaluate, monitor, and improve global climate policy.

5.2 Data and Labeling

As raw data, 165 English-language NDCs and Intended NDCs [123] were obtained in HTML format from Climate Watch [3]. Paragraph, list, and table elements were extracted from these HTML documents. The text elements typically contain multiple sentences, sentence fragments, and in the case of tables, numeric data. Numeric data were removed from the tables, and text elements were sentencized [48]. Sentences under 10 words in length were discarded to remove less useful sentence fragments. This process generated 25,500 unique sentences. Document length ranged from 18 to 482 sentences with a mean of 154 and a median of 130 sentences. Manual classification of sentences into topic groups was not feasible. Instead, "weak" labels were generated for each sentence by exploiting the nested headers, subheaders, and table structures within the HTML documents.

The full set of lemmatized words found in the HTML headers and table row names were manually divided into 11 topic areas by human climate policy experts. For example, terms such as "deforestation" or "LULUCF" (land use, land-use change, and forestry) were assigned to the Land Use topic. The topic identified with the most deeply nested header was assigned to all sentences within that text element. In cases where multiple topic words appeared in a given header, the less frequent topic was assigned (e.g., if both Land Use and Mitigation keywords were present then the less frequent Land Use label was applied). In some cases, no topic was assigned in which case the sentence was labeled as "No Label." The distribution of topics was not uniform. Some topics, such as Mitigation or Adaptation appear more frequently than others such as Industry or Environment (Table 1). These reference labels are referred to as weak labels to emphasize that they are noisy and often do not correspond to topic labels that would be assigned by human annotators.

Table 5.1: Frequency of weak labels over NDC sentences.

WEAK LABEL	FREQUENCY (%)
NO LABEL	16.3
ADAPTATION	15.0
AGRICULTURE	4.7
ECONOMIC	4.5
ENERGY	5.0
ENVIRONMENT	3.0
EQUITY	7.1
INDUSTRY	2.0
LAND USE	3.4
MITIGATION	16.0
STRATEGY	21.7
WASTE	1.2

5.3 Model Framework

The weakly labeled sentences were split into training, validation, and test sets comprising 80, 10, and 10 percent of the sentences, respectively. The transformer models were iteratively optimized on the training data. At each epoch of the training process, model loss was calculated using the validation data. If the validation loss increased for three epochs in a row the training process was halted and the model with the lowest validation loss was chosen. The final model was then applied to the hold-out test set of sentences to evaluate model performance.

Two uncased transformer models were trained and tested against the data: $\text{BERT}_{\text{BASE}}$ and SciBERT [11]. SciBERT is a BERT model pretrained on a large corpus of scientific publications which has been shown to provide improvements on standard NLP tasks on data sets from scientific domains. Examples of the training, validation, and test set evaluation scores are shown for the BERT model in Table 2.

Table 5.2: BERT evaluation metrics. Precision and recall calculated using macro averaging.

DATA	ACCURACY	PRECISION	RECALL	F_1
TRAIN	0.907	0.692	0.673	0.669
VALIDATE	0.839	0.450	0.436	0.429
TEST	0.847	0.417	0.406	0.397

The BERT and SciBERT model performances were compared to three benchmark classifiers. The null Random classifier assigned labels randomly with equal frequencies. The Majority classifier assigned the most common topic, Strategy, to all sentences. The Contains classifier applied a simple heuristic: if any of the topic words associated with a topic label appeared in a sentence then the sentence received that topic label. As with the weak reference labels, if multiple topic words appeared within the same sentence the lowest frequency topic label was assigned.

The reasoning is that lower frequency labels have greater specificity and are likely to capture more salient content.

Finally, a balanced 600-sentence subset of the test set (50 sentences with each weak label) was manually labeled by two student annotators. The human labels were evaluated relative to the weak labels to provide an upper bound to the NLP classification metrics. The two sets of human labels were compared to quantify inter-annotator agreement.

5.4 Results

5.4.1 Model Evaluation

Table 3 presents weighted evaluation metrics for each of the classifiers. The Random and Majority classifiers perform poorly with weighted F_1 scores of 0.09 and 0.07, respectively. The Contains heuristic shows some improvement over the null classifiers with F_1 of 0.17. BERT outperforms these classifiers with F_1 of 0.40. SciBERT is marginally less accurate than BERT and has a lower F_1 score, perhaps indicating that the policy documents are more similar to general text corpora than to collections of scientific documents.

Table 5.3: Model performance. Precision and recall calculated using macro averaging.

CLASSIFIER	ACCURACY	PRECISION	RECALL	F_1
RANDOM	0.813	0.117	0.081	0.089
MAJORITY	0.757	0.041	0.203	0.069
CONTAINS	0.829	0.229	0.170	0.171
SCIBERT	0.843	0.398	0.379	0.362
BERT	0.847	0.417	0.406	0.397
HUMAN*	0.867	0.281	0.250	0.251

* The Human metrics are calculated on a 600-sentence subset of the hold-out test set.

To put the BERT F_1 score in context, the Contains and BERT predicted labels were tested against the human labels (Table 4) on the balanced 600-sentence subset of the test set. One of the human annotators, referred to here as Student, was a student researcher with no knowledge of climate policy who was simply directed to label sentences using their best judgment. The other human annotator, the Expert, was a student with climate policy research experience and familiarity with the NDC documents. In these results "Human" scores are averages over both annotator scores.

On average, the simple Contains heuristic shows better agreement with the human annotators' labels than the BERT classifier. This is not surprising, given that BERT was optimized to predict the weak labels which provide a very noisy representation of semantic content. Ideally BERT would be trained on human-annotated data, but in many applications such data sets are expensive to generate.

Interestingly, the Contains heuristic only outperforms BERT trained on weak labels when compared with the Student labels. When compared to the Expert labels BERT slightly outperforms

Table 5.4: Comparison of Contains and BERT to human annotators.

WEIGHTED F_1		REFERENCE LABEL		
		HUMAN	STUDENT	EXPERT
CLASSIFIER	CONTAINS	0.350	0.399	0.302
	BERT	0.301	0.284	0.317
	STUDENT			0.472

Contains although the difference in F_1 scores is not significant (using bootstrapped 95% confidence intervals). It may be that the Contains heuristic more closely mimics an untrained annotator while BERT is better able to emulate expert-level context-sensitive classification. More annotated data would be required to explore this possibility.

5.4.2 Error Analysis

An illustrative set of test sentences (edited here for concision) are presented in Table 5 with their weak reference labels and the Contains, BERT, Student and Expert predicted labels. In the first sentence the classifiers agree: the keywords "emission" and "mitigation" both indicate that the sentence concerns Mitigation. The second sentence is correctly labeled Strategy by BERT and the human annotators, but not by Contains, i.e., BERT outperforms Contains. The third sentence contains keywords from different topics ("sequestration" indicates the Mitigation topic; "afforestation" indicates Land Use). Here Contains matches the weak label. BERT predicts Agriculture which is semantically similar to Land Use suggesting potential improvements to the classification algorithm. The Student seizes upon the first keyword "mitigation" as a label, demonstrating a potential weakness of manual annotation. The fourth sentence predictions are confused although BERT has matched the Expert annotator's label. Finally, the last sentence is not related to climate change but instead provides background information on a country's recent history. Relative to manual annotation the weak reference label seems inappropriate; in this case the sentence has fallen under an HTML section header that indicates the Environment topic. It is not surprising that none of the classifiers match the weak label.

5.5 Discussion and Future Work

Using weak topic labels derived from the document header structure as reference labels is clearly inferior to a system in which a large number of training, validation, and test sentences are manually annotated by climate policy experts. However, manual annotation is often infeasible for large corpora. While BERT outperforms simpler methods and even the human annotators in predicting the weak reference labels, the simpler Contains classifier provides better agreement with the human annotators. The difference is not as pronounced when the annotator with more domain-specific expertise is used as the reference, but the results underscore the importance of clean reference data for training deep learning models.

The development of this framework suggests several areas for improvement. First, sentences could be classified with multiple labels. For example, the phrase "The mitigation actions that

Table 5.5: Error analysis.

SENTENCE	LABEL	CONTAINS	BERT	STUDENT	EXPERT
It is envisaged that **emission** reduction will be achieved through the **mitigation** actions in the sectors.	MITIGATION	MITIGATION	MITIGATION	MITIGATION	MITIGATION
The Steering Committee is the supreme body for **decision making** and sectoral **implementation**.	STRATEGY	MITIGATION	STRATEGY	STRATEGY	STRATEGY
The **mitigation** actions that enhance **afforestation** are projected to result in the **sequestration** of 1 mtCO2e annually.	LAND USE	LAND USE	AGRICULTURE	MITIGATION	LAND USE
In the absence of project activity, **fossil fuels** could be burned in **power plants** that are connected to the grid.	STRATEGY	EQUITY	ENERGY	INDUSTRY	ENERGY
Due to the outbreak of the Ebola Virus the development gains made after a 10-year civil war were rudely reversed.	ENVIRONMENT	NO LABEL	MITIGATION	NO LABEL	NO LABEL

enhance afforestation are projected to result in the sequestration of 1 mtCO2e annually" could be classified as both a Mitigation sentence and a Land Use sentence.

To generate predicted topic labels, BERT takes the argmax of a vector of weights derived from the final hidden layer of the neural network. These weights can be normalized to provide sentence topic probabilities. Such soft labels may generate more meaningful document-level climate policy metrics than hard single-topic sentence labels. Similarly, simple bag-of-words classifiers often yield multiple predictions when conflicting keywords appear in the same sentence. In this study, the two human annotators were instructed to list all relevant topics and then to choose the topic they felt was most relevant. The next step in this research will be to evaluate these multi-label classifiers against weak multi-labels and against each other.

A more complex multi-labeling approach could account for hierarchies within the set of topics. For example, there are energy strategies that fall under the framework of mitigation (e.g., transition to renewables) and energy strategies that fall under the framework of adaptation (e.g., protection of nuclear power plants from sea level rise). Classification with hierarchical topic labels could further improve metrics for policy analysis.

The selection of topics and of topic words using text from the HTML headers was performed by two trained climate researchers. Manual classification is inherently subjective, and compromises were required between the experts to obtain a reasonable classification scheme. Automated approaches such as those discussed in Lucioni & Palacios [76] or Kölbel et al. [56] may offer some improvement. Furthermore, the number of topics selected in this analysis was limited to 11 plus the null label for ease of computation and to avoid problems arising from sparse labels. Initially over 25 topics were proposed on the basis of the header words. If possible, it would be interesting to extend the analysis to consider a much wider range of topics, including specialized topics such as indigenous community involvement [31] or the impacts of climate change on

coastal communities and marine ecosystems [42].

5.6 Conclusion

Under the Paris Agreement, signatories are expected to submit updated NDCs every five years. As of May 2021, eight countries have submitted their second NDC [123] though more plans are expected once the COVID-19 pandemic is brought under control. In the U.S., 33 states have released climate action plans [21]. Globally, 28 cities in the C40 Cities Climate Leadership Group have published Paris Agreement compatible climate action plans [22]. The continued development of state-of-the-art NLP tools tailored to climate policy will allow climate researchers and policy makers to extract meaningful information from this growing body of text, to monitor trends over time and administrative units, and to identify potential policy improvements.

Chapter 6

Towards Answering Climate Questionnaires from Unstructured Climate Reports

- Daniel Spokoyny and Tanmay Laud and Tom Corringham and Taylor Berg-Kirkpatrick, "Towards Answering Climate Questionnaires from Unstructured Climate Reports.", In Proceedings of EMNLP Workshop on NLP for Positive Impact 2022

6.1 Introduction

As mentioned earlier, there is an evergrowing body of climate reports generated by different *stakeholders* such as corporations, cities, states, and national governments either voluntarily or in response to regulatory pressure. These reports disclose vital information on carbon emissions, impacts, and risks – for example, a firm's emissions reduction targets or a city's water risk and exposure to drought. Increasingly, NLP is a critical technology supporting large scale processing of climate reports to enable downstream applications like detecting corporate greenwashing [18] or identifying misinformation about climate change [79]. However, for climate researchers to make use of the information contained in these *unstructured* text documents, their contents must first be collated into *semi-structured* questionnaires that have consistent fields across reporting bodies and report types. These structured questionnaires, in turn, allow climate researchers to compare progress across different stakeholders and identify which areas need financing, education, policy changes or other resources. Currently, this extraction process requires an immense amount of manual effort resulting in whole organizations focused on mapping a single type of unstructured reports (Nationally Determined Contribution) to a single type of semi-structured questionnaires (Sustainable Development Goals).[1][2]

In order to facilitate NLP research for this task, we introduce two new datasets, CLIMA-CDP and CLIMA-INS, which are composed of publicly accessible semi-structured questionnaires

[1]World Resources Institute's: www.climatewatchdata.org

[2]For more background info see Appendix 6.9.

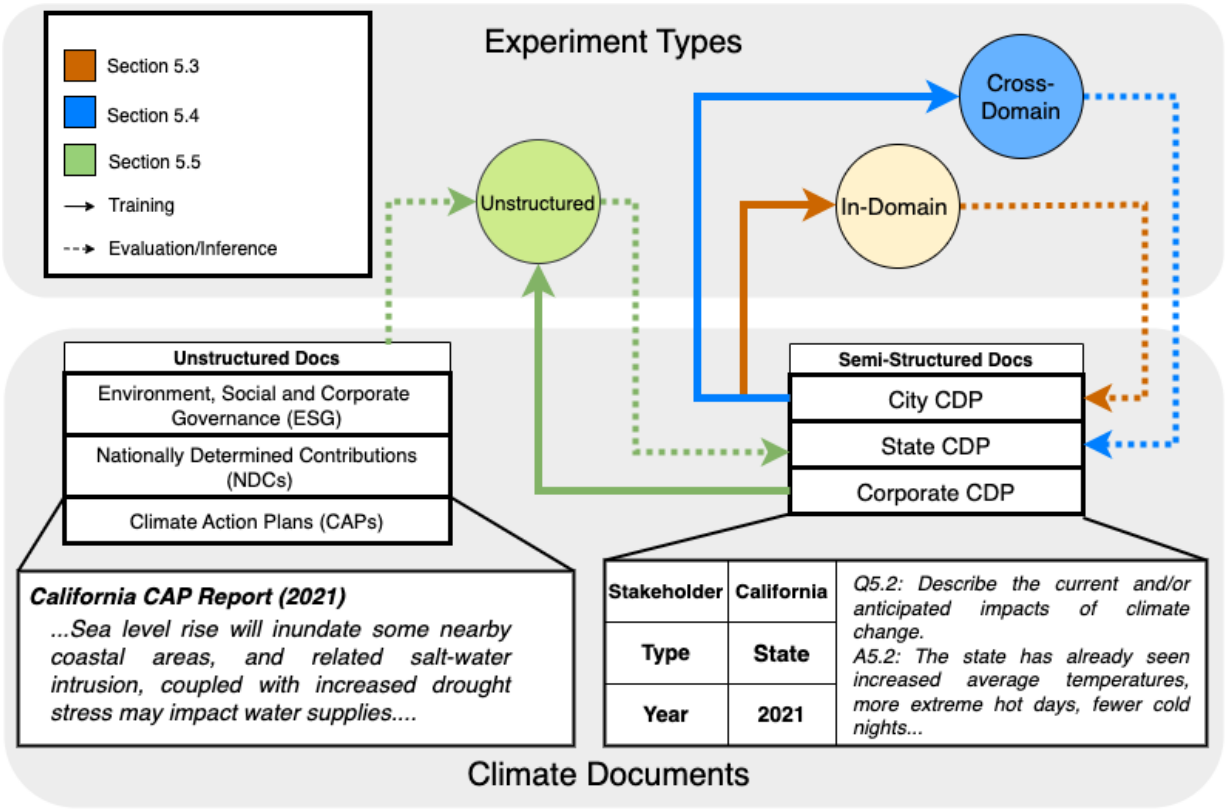


Figure 6.1: We conduct 3 experiments on CDP-QA. In-Domain (6.5.3) refers to training and evaluating on the same *stakeholder-type*. Cross-Domain (6.5.4) refers to training and testing on different *stakeholder-type*s. Finally, Unstructured Questionnaire Filling (6.5.5) involves training on the whole CDP-QA corpus and then using the model for mapping text from a CAP report to a CDP. We use solid and dashed arrows to denote training and inference/evaluation respectively.

from different stakeholders including cities, states and corporations. We utilize the structure of the questionnaires to train self-supervised classification models to align answers to questions (Section 6.5.3). Further, we show how the setup of our objective allows our model to generalize to a more challenging scenario where the set of questions and the stakeholder-type are both different at test time (Section 6.5.4). Finally, we show that models trained on CLIMA-CDP can be directly applied to map passages from unstructured documents into questionnaire categories, which matches the real-world use-case that climate researchers need solved (Section 6.5.5). In Figure 6.1 we depict all three of these experiments as well as examples of the different reports and stakeholder-types.

There are other existing climate-specific datasets for detecting relevance to climate [64], identifying stance detection [125] and fact-checking [63] of social media claims. In contrast, the questionnaires we introduce have an order of magnitude more data, are comprehensive in both the breadth of topics covered and the depth of detail provided making our models most suitable for a wide range of climate applications.

Climate reports have also been used as a source of unlabeled data to continue pretraining large language models to better adapt them for climate specific tasks [77, 135]. However, it remains an open question whether these domain-specific models can effectively generalize since evaluation of these models has been limited on the climate domain. To address this gap in comprehensive evaluation, we collate five existing climate datasets, along with our two new datasets into a benchmark dataset (CLIMABENCH), and find that the domain-specific models like ClimateBERT underperform compared to existing general models (Section 6.5.2).

In summary, our contributions are as follows:

1. We introduce two new datasets, CLIMA-CDP and CLIMA-INS, consisting of difficult classification tasks that are analogous to current manual work done by climate researchers, and conduct extensive in-domain experiments.
2. We collate and release CLIMABENCH, an evaluation dataset of climate-related text classification tasks and show that, counter-intuitively, general-purpose ML models outperform domain-specific models across tasks within the benchmark.
3. We conduct a pilot study, evaluated manually by a climate researcher, that uses a model trained on CLIMA-CDP to populate a questionnaire from real-world unstructured climate reports.

We believe our contributions are an important step for an emerging domain of building NLP tools for climate researchers. To that end, we release our benchmark[3] and open-source our trained models[4] to encourage researchers to extend our existing datasets and contribute new ones.

6.2 Related Work

Climate policy evaluation is an active area of research in climate sciences where the goal is to evaluate the effectiveness of current climate policies so as to inform future policy decisions [25, 113]. It allows for the development, assessment, and improvement of regulation, increases

[3] https://github.com/climabench/climabench

[4] https://huggingface.co/climabench/miniLM-cdp-all

transparency and public support, and encourages public and private sector entities to make pledges or increase their levels of action [41, 96]. NLP has the potential to derive understandable insights from policy texts for these applications.

Academic literature provides a valuable source of information for conducting these evaluation studies. However, a necessary first step is systematic evidence mapping or identifying which papers are relevant to a particular policy. Berrang-Ford et al. [14], for instance, build a machine learning system to filter scientific literature relevant to climate adaptation.

Another area of research involves utilizing unstructured climate documents for topic classification. Corringham et al. [30] attempt to use document headers from unstructured Nationally Determined Contribution (NDC) reports as coarse-grained labels to train a supervised classifier. Most similar to our CLIMA-QA work is Luccioni et al. [77] who trained a model to map text passages from public financial disclosures to the 14 questions in Task Force on Climate-related Financial Disclosures (TCFD). They recruited experts to manually label the text passages to the TCFD questions and only train their models on this labeled data. Our work focuses on using the existing structure of large-scale public questionnaires to first train models and then apply them to unstructured texts.

NLP is also used to analyze social media data to understand public opinions and discourse around climate change [55]. CLIMATEXT [64] and CLIMATEFEVER [63] extracted and filtered documents from Wikipedia and other sources to curate a CC corpus that was further annotated by humans. In climate finance, Kölbel et al. [57] have built NLP classifiers to distinguish texts describing physical climate risk versus transition risk. While these studies have independently analyzed small annotated datasets, we make use of semi-structured disclosure forms comprising a much larger set of supervised data, made available to the CC and NLP communities in a clean and accessible format. Similar work has been conducted manually in the CC policy evaluation community (e.g., ClimateWatch) but not over the breadth and scope of documents we consider.

Finally, benchmarks have been an effective way to track progress and highlight the shortcomings of NLP models in both general-purpose understanding (GLUE [132], SuperGLUE [131]) as well as specific domains such as legal NLP (LexGLUE [27]) or biomedical NLP (BLURB [46]). CLIMABENCH follows on this chain of thought to provide a unified way to evaluate models on CC-specific problems.

6.3 Datasets

In this section we first describe our two new questionnaire datasets, CLIMA-CDP and CLIMA-INS, and then present all the text classification datasets we collected into CLIMABENCH. We consider a questionnaire, a semi-structured document, filled out by a *stakeholder* for a particular year to have a set of questions and answers, (Q, A) where the i-th question-answer pair $\{q_i, a_i\}$ are both free-form text. Table 6.1 lists a few interesting examples from the newly introduced datasets. The overall statistics of each dataset are given in Table 6.2, the token length distribution is given in Appendix Table 6.8 and details are explained below.

	Free-form Text/Answer	Class / Question	# Classes
CLIMA-INS	...Each year Aflac reports its US operations Scope 1 and Scope 2 emissions to the Carbon Disclosure Project. Since 2007, Aflac's owned facilities in terms of square feet have increased by more than 10% while total Scope 1 and 2 CO2e emissions have significantly decreased compared to 2007 emissions...	Does the company have a plan to assess, reduce or mitigate its emissions in its operations or organizations?	8
CDP-TOPIC	...These Plans must include management of CD&E waste, both through on-site recycling and re-use and on-site waste processing prior to disposal.Westminster will contribute to the London Plan target of net self-sufficiency (managing 100% of London's waste within London) by 2026 by planning for Westminster's apportionment targets...	Governance and Data Management	12
CDP-QA (Cities)	Flooding from sea level rise will damage building and roads in the coastal neighborhoods of the city. Flooding also represents a risk to major transportation hubs infrastructure in the region. Coastal flooding can have a long-term effect on major industrial and commercial activities along the coastal areas of the city as well as damage urban forestry and local natural biodiversity.	Please describe the impacts experienced so far, and how you expect the hazard to impact in the future.	294

Table 6.1: Examples (pairs of inputs and outputs) for the newly introduced datasets.

Dataset	Source	Task Type	Domain	Stakeholder	# Train	# Dev	# Test	# Classes
CLIMA-INS	Ours	Multi-class Classification	NAIC	Corporations	13.7K	1.7K	1.7K	8
CDP-TOPIC	Ours	Topic Classification	CDP	Cities	46.8K	8.7K	8.9K	12
CDP-QA	Ours	Question Answering	CDP	Cities	48.2K	8.5K	9.3K	294
				States	8.7K	0.9K	1.1 K	132
				Corporations	34.5K	3.6K	4.9K	43
CLIMATEXT	Leippold and Varini [64]	Binary Classification	Wikipedia, 10-K	-	6K	0.3K	1.6K	2
CLIMATESTANCE	Vaid et al. [126]	Ternary Classification	Twitter	-	3.0 K	0.3K	0.3K	3
CLIMATEENG	Vaid et al. [126]	Multi-class Classification	Twitter	-	3K	0.3K	0.3K	5
CLIMATEFEVER	Leippold and Diggelmann [63]	Fact-Checking	Wikipedia	-	-	-	1.5K	3
SCIDCC	Mishra and Mittal [81]	Topic Classification	Science Daily	-	9.2K	1.1K	1.1K	20

Table 6.2: General statistics of the datasets collected for CLIMABENCH and CDP-QA.

6.3.1 CLIMA-INS

The annual NAIC Climate Risk Disclosure Survey[5] is a U.S. insurance regulation tool where insurers file non-confidential disclosures of their assessments and management of climate-related risks. The purpose of the survey is to enhance transparency about how insurers manage climate-related risks and opportunities to enable better-informed collaboration on climate-related issues. The dataset contains survey responses for the years 2012-2021, where each survey consists of eight questions all shown in Appendix 6.8 and examples in Table 6.1. Companies have an option to fill the survey individually or as a group (in case of a conglomerate). In the case of group filing, there may be duplicate answers repeated across all subsidiaries. We remove such responses resulting in a total of 17K question-answer pairs. Further, we delete the first sentence in each response as it contains obvious markers (like "Yes, we do X." or "No, we do not participate in Y."). The splits for training, validation and testing (80%, 10%, 10%) are created by stratifying based on the company so that similar responses from the same company are not seen during train and test.

[5]https://interactive.web.insurance.ca.gov

6.3.2 CLIMA-CDP

Carbon Disclosure Project (CDP) is an international organisation that runs a global disclosure questionnaire for various stakeholders to report their environmental information. In 2021 alone over 14,000 organizations filled out the questionnaire which contains hundreds of unique questions.

The CLIMA-CDP, D_{cdp}, is composed of parts [D_{city}, D_{corp}, D_{state}] where each part is a set of questionnaires filled out by a city, company, or state respectively. From the questionnaires we construct two tasks: topic classification (CDP-TOPIC) and question classification (CDP-QA).

CDP-TOPIC The CDP questionnaire contains a hierarchy of questions organized by topics such as *energy*, *food*, *waste*. We utilize these topics as *labels* for a classification task and show the mapping in Appendix Table 6.7. Thus, for each question-answer pair $\{q_i, a_i\}$ we also have a topic label. We formulate a topic classification task where the goal is to predict the *topic* given the text of the *answer*.

CDP-QA Our aim is to construct controlled experiments with proper evaluation metrics which closely resemble the real-world scenario of aligning unstructured climate reports to semi-structured ones. For example, the CDP DATASET allows us to test whether models can generalize to questionnaires of different *stakeholder-type*. However, since the set of questions for each stakeholder type (Q_{city}, Q_{corp}, Q_{state}) are different from one another, a classifier predicting the question type will not be able to transfer to a new stakeholder type. By using the text of the questions directly we can handle new questions at test time, which allows us to train on questionnaires from cities and test their generalization on questionnaires for states. Since organization may file yearly reports which contain similar information we build train, dev and test splits stratified by the organizations. Further we filter out duplicate, non-English, and short (less than 10 words) responses.

6.3.3 CLIMABENCH

In this section we introduce CLIMABENCH, a benchmark of climate related text classification tasks for evaluating NLP models. We collate five existing climate change related text datasets, described in detail below along with CLIMA-INS and CDP-TOPIC.

CLIMATEXT is a dataset for sentence-based climate change topic detection [64]. Each sentence is labelled indicating whether it is relevant to climate change or not. Sentences were collected from the general web and Wikipedia as well as the climate-related risks section of US public companies' 10-K reports.

CLIMATESTANCE and **CLIMATEENG** Vaid et al. [126] extracted Twitter data consisting of 3777 tweets posted during the 2019 United Nations Framework Convention on Climate Change. Each tweet was labelled for two tasks: stance detection and categorical classification. For the stance detection the authors labelled each tweet as *In Favour*, *Against* or *Ambiguous* towards climate change prevention. For categorical classification, the five classes are *Disaster*, *Ocean/Water*, *Agriculture/Forestry*, *Politics*, and *General*.

CLIMATEFEVER [63] adopts the FEVER [119] format for a fact-verification task based on climate change claims found on the Internet. The dataset consists of 1,535 claims and five relevant evidence passages from Wikipedia for each claim. The label set for each claim-evidence pair is *Supports*, *Refutes*, or *Not Enough Info* for a total 7675 labelled examples. For CLIMATEFEVER,

we concatenate the texts of each claim-evidence pair as a single input to the model.

SCIDCC [81] The Science Daily Climate Change or SCIDCC dataset is curated by scraping news articles from the Science Daily website [81]. It contains around 11k news articles with 20 labeled categories relevant to climate change such as *Earthquakes*, *Pollution* and *Hurricanes*. Each article comprises of a title, a summary, and a body which on average is much longer (500-600 words) than the other climate text datasets. For SCIDCC, we concatenate the text fields (title, summary and body) and provide a train, validation and test split (80%, 10%, 10%) for this data, ensuring the distribution of categories in the splits matches the overall distribution.

6.4 Models

Next, we are going to describe the various baselines and models that we use to conduct experiments using the datasets described above. Most tasks are classification tasks that require in-domain finetuning. For the text classification tasks in CLIMABENCH, we examine Transformer-based [128] pre-trained language models like BERT [35], RoBERTa [71], distilled versions like DistilRoBERTa [102], longer context models like Longformer [13], and domain specific models like ClimateBERT [135] and SciBERT [12]. This helps us contrast the effects of model architecture, input length and in-domain pretraining on downstream tasks. We provide more details about models in Appendix Section 6.8.3 and Table 6.9. For a baseline, we consider a linear kernel Support Vector Machine (SVM) trained using TF-IDF transformed n-gram (1,2,3-gram) features. We also include a simple Majority and Random class voting baselines.

For experimentation on CDP-QA we consider a pre-trained Cross-Encoder MiniLM [133] model which was separately finetuned on the MS MARCO Passage Retrieval Dataset [23] by Reimers and Gurevych [92]. The MS MARCO dataset contains real user queries together with annotated relevant text passages. The model takes in as input the query concatenated with the passage and is trained to predict the pair's binary relevance score. This model achieved state of the art performance across many retrieval tasks [114]. We consider this as a strong general purpose model in contrast to ClimateBERT which is a domain specific model.

6.5 Experiments

In our work we conduct four experiments: **(1)** climabench classification, **(2)** in-domain self-supervised questionnaire filling, **(3)** cross-domain questionnaire filling, and **(4)** unstructured questionnaire filling. For the first experiment, we examine the performance of existing general models as well as climate-specific models on our new CLIMABENCH evaluation dataset. Experiments 2 and 3 focus on how we can utilize the semi-structured CLIMA-QA dataset to create a self-supervised version of the unstructured document alignment task in a controlled setting with proper evaluation metrics. Finally, in experiment 4 we will evaluate using human relevance judgements a model trained using the semi-structured CDP dataset can aid in aligning an unstructured climate report to the CDP questionnaire.

Models	CLIMA-INS	CDP TOPIC	CLIMA-TEXT	CLIMATE-STANCE	CLIMATE-ENG	SCIDCC	CLIMATE-FEVER	AVG.
Majority	4.11	3.65	42.08	29.68	13.83	0.79	26.08	20.10
Random	12.14	6.45	46.86	25.52	16.71	5.05	30.62	24.09
SVM	**86.00**	58.34	83.39	42.92	51.81	48.02	-	-
BERT	84.57	64.64†	87.04†	55.37†	71.78	54.74†	62.47†	70.57†
RoBERTa	85.61†	**65.22**	85.97	**59.69**	**74.58**	52.90	60.74	**71.14**
DistilRoBERTa	84.38	63.61	86.06	52.51	72.33†	51.13	61.54	69.27
Longformer	84.35	64.03	**87.80**	34.68	72.28	**54.79**	60.82	67.72
SciBERT	84.43	63.62	83.29	48.67	70.50	51.83	**62.68**	68.45
ClimateBERT	84.80	64.24	85.14	52.84	71.83	52.97	61.54	69.44

Table 6.3: Macro F1 Scores on the Classification Datasets. **Bold** and † indicate first and second highest performing model respectively. RoBERTa scores the best on average followed by BERT and ClimateBERT.

6.5.1 Task Learning Details

Each task has its own supervised training data that allows for in-domain finetuning for the target classification task. In all experiments for all transformer models except MiniLM, we will add a classification head and do full finetuning. For all the pre-trained models, we use publicly available Hugging Face [141] checkpoints.[6] For the Longformer, we use the default settings (windows of 512 tokens and a single global $[CLS]$ token). We use the Scikit-learn API [87] for the simple classifiers (Random and Majority class) and TF-IDF-based linear SVM models. We grid-search the hyper parameters for SVM with 5-fold validation (Table 6.11).

We use a training batch size of 32 and optimize using AdamW [74] with a learning rate of 5e-5 (linear warm-up ratio of 0.1, weight decay of 0.01) for 10 epochs with early stopping based on performance on development data (F1). We use mixed precision (fp16), gradient checkpointing and gradient accumulation steps of 2 to train models efficiently on the limited compute (Appendix 6.8.1). We truncate the input text when it exceeds the maximum input length of the model and otherwise pad the input.

6.5.2 Text Classification on CLIMABENCH

In this section we use the new text classification CLIMABENCH dataset as an evaluation framework to compare the performance of the different models. We use macro-averaged F1 score as our

[6]We use the *-base configuration of each pre-trained model, i.e., 12 Transformer blocks, 768 hidden units, and 12 attention heads. For ClimateBERT we report scores for the F variant model on Huggingface. For the QA Cross-encoder, we use the MiniLM (12 layer, 384 hidden-unit) finetuned on MSMARCO available at `https://huggingface.co/cross-encoder/ms-marco-MiniLM-L-12-v2`

evaluation metric since the datasets are imbalanced and all classes are equally important. For the pre-trained transformer models, we add a single linear classification layer on top of the final $[CLS]$ token representation and use a weighted cross-entropy loss with class balanced weights.[7]

Results on CLIMABENCH

We report text classification results on CLIMABENCH in Table 6.3 as well as an average across all tasks. We find there is no single model that does the best across the board, but RoBERTa is a clear winner as it beats the other baselines on four out of eight tasks. Both of the domain adapted models, SciBERT and ClimateBERT do worse than their non-adapted counterparts. For example, ClimateBERT and the model it was warm-started from, DistilRoBERTa, are very similar in performance. Overall, the transformer models have significantly better gains over linear ones except on CLIMA-INS where the TF-IDF+SVM model is superior. It shows that simple word co-occurrence statistics are enough for certain tasks and deep language models might not be the right solution in such cases.

6.5.3 In-Domain CDP-QA

	CDP-CITIES	CDP-STATES	CDP-CORP
Model	**MRR@10**	**MRR@10**	**MRR@10**
	No Finetuning on CDP		
BM25	0.055	0.084	0.153
MiniLM	0.099	0.120	0.320
	Finetuned on CDP		
	In-Domain	In-Domain	In-Domain
ClimateBERT	0.331	0.422	0.753
MiniLM	**0.366**	0.482	**0.755**
	Best Model Finetuned on all		
MiniLM	0.352	**0.489**	0.745

Table 6.4: MRR@10 scores for BM25, ClimateBERT and MSMARCO-MiniLM on the three subsets of CLIMA-QA. Models finetuned and evaluated on same subset fall under In-Domain.

We utilize the semi-structured nature of the questionnaire to train models in self-supervised fashion. Specifically, we concatenate the free-form text of the answer and question and train a

[7]We do not evaluate linear models on fact-checking or QA as the heterogeneity of the input in these tasks do not align with the linear setup.

binary classifier to predict whether, in fact, the input answer matches the input question – i.e. does a_i, the ith answer in our dataset, provide an answer to q_j, the jth question in our dataset: $p(y_{ij} = 1|q_j, a_i)$. Since we assume that the indices are setup so that a_i matches q_j if and only if $i = j$, the ground truth labels are given by $y^*_{ij} = \mathbb{1}[i = j]$.

We use the filled out questionnaires as positive or relevant pairs and randomly sample five negative QA pairs for each relevant pair. We train separate models on each *stakeholder-type* partition of the CDP DATASET and evaluate them on the corresponding in-domain test sets. During inference time, given an answer we compute a relevance score for all combinations of QA pairs from the full set of questions of a particular *stakeholder-type*.

$$\text{argmax}_{j \in \{1,\ldots,|Q|\}} \; p(y_{ij} = 1|q_j, a_i)$$

Since there is a large number of questions, instead of accuracy we consider the Mean Reciprocal Rank at k (MRR@k) scores for the top k items returned by a model. MRR, a popular metric used in the Information Retrieval field, is the average of the reciprocal ranks of results for a sample of queries where the relevance grading is binary (Yes/No).

We narrow down to two models, MiniLM and the ClimateBERT model to study the effects of fine-tuning and transfer learning on the three subdomains: CDP-CITIES, CDP-STATES and CDP-CORP. We also use BM25 [93] and MiniLM with no training as baselines.

Results

We report the results of our in-domain experiments on CLIMA-QA in Table 6.4 (detailed results in Appendix Table 6.12). We find that MiniLM, a much smaller model, beats ClimateBERT across all three different subsets. It is hard to diagnose the exact reason why domain adaptation does not help in this case as well since the data used to further pretrain ClimateBERT is non-public. There may be further room for improvement in domain adaptation for the MiniLM, but we leave this as future work. Lastly, the best performing model, MiniLM, when finetuned on all three subsets, achieves comparable performance on Cities and Corporations while ranking highest on States.

6.5.4 Transfer CDP-QA

In this section we explore whether it is possible for transfer learning to adapt to questionnaire from a new unseen *stakeholder-type*. Since the D_{city} dataset is the largest we use this partition as our training data. Furthermore, since we have the ground truth questionnaires for both states and corporations we are able to evaluate the performance in a controlled setting. At test time we follow the same procedure as for the in-domain experiment however, we marginalize over the set of questions from the unseen *stakeholder-type*.

Results

We summarize the MRR@k (k=10) results for the transfer learning experiments in Table 6.5 (detailed results in Appendix Table 6.13). We show that both models are able to beat the no-training baselines. We again find that the MiniLM model outperforms the ClimateBERT model across both transfer learning scenarios. We do observe a significant drop in performance as compared to

	CDP-STATES	CDP-CORP
Model	**MRR@10**	**MRR@10**
	No Finetuning	
BM25	0.084	0.153
MiniLM	0.120	0.320
	Finetuned on CDP-CITIES	
	Transfer	Transfer
ClimateBERT	0.298	0.465
MiniLM	**0.353**	**0.489**

Table 6.5: MRR@10 scores for BM25, ClimateBERT and MiniLM on the Transfer experiments. Models are finetuned on CDP-CITIES and evaluated on States and Corporations.

	Prec@1	Prec@2	Prec@3	Prec@4	Prec@5
Relevant	63.0	67.0	68.6	69.5	71.0
Highly Relevant	30.0	32.0	32.3	32.5	35.6

Table 6.6: Precision@K: We report the fraction of items in the top K ranked retrievals that are either marked as highly relevant, or at least relevant, averaged across text examples. Relevance judgements were performed manually by an expert annotator.

the in-domain finetuning experiments. This gap is the largest for the corporations dataset, where the MRR@10 drops from 0.745 to 0.48. Overall, we find that the transfer learning models are able to adapt to the unseen *stakeholder-type* but that there is still room for improvement.

6.5.5 Questionnaire Filling

In our final experiment we consider the task of filling in a questionnaire based on an *unstructured* text document – specifically, we assume a State's Climate Action Plan (CAP) is available but the corresponding structured CDP report is not. Typically the CAPs are much longer ($\sim$100 pages) and more comprehensive than any particular disclosure report. The CAPs include quantitative data, such as emission values or renewable electricity generation capacity, and qualitative data such as specific policy interventions across different sectors. Populating CDP questionnaires allows for consistent comparisons to existing datasets which could further be used to compare strategies, identify gaps, or rank jurisdictions on the content and level of ambition in their stated plans. However, this process is time-consuming and requires expert manual effort.

We select our best model, MiniLM, finetuned on the full CLIMA-CDP dataset to conduct our unstructured questionnaire filling. We can consider a State CAP as an unstructured document D_{un}, to be a collection of texts, $D_{un} = \{t_1, t_2, \ldots, t_n\}$, where t_i is a text segment. The task is then to align a text segment t_i to its corresponding CDP-State question $q_j \in Q_{state}$, i.e. $\text{argmax}_{j \in \{1,\ldots,|Q_{state}|\}}\ p(y_{ij} = 1 | q_j, a_i)$. Since we do not have the ground truth alignment we use a climate change researcher in a procedure as follows: 1) First, the expert (climate policy researcher

on our team and co-author) selected 5 pages at random from a collection of 20 State CAPs and then selected a random paragraph from each page as a text segment t_i. 2) Then, using our model we calculated relevance scores for each text segment question pair (t_i, q_j) and selected the top 5 scoring questions for each text segment. 3) We then presented each segment along with the five questions to the climate change researcher and had them annotate the relevance for each pair on a three point scale: No Relevance, Relevant, Highly Relevant.[8]

Human Evaluation

Table 6.6 shows the climate change researcher's evaluation metrics for our model. Overall, 71.0% of the 500 questions retrieved were judged *Relevant* and 35.6% rated *Highly Relevant*. One pitfall of our model is that there were more very relevant predictions ranked fifth than first. One possible explanation for this is that the top retrieved questions were often more general while the questions that were ranked lower were more specific and easier to match (see Table 6.14 in the Appendix). We show some examples of text segments and the selected questions in Appendix Table 6.15. Although our pilot study is quite limited, it shows both the promise and the challenges of aligning unstructured climate documents to semi-structured questionnaires.

6.6 Conclusion

In summary, we introduced two climate questionnaire datasets and illustrated how using their existing structure we can train self-supervised models for climate question answering tasks analogous to real-world challenges faced by climate researchers. Finally we lay the groundwork for future work in this domain by introducing a collated benchmark of existing climate text classification datasets.

6.7 Limitations

One current limitation of our benchmark is that the datasets are English only, thus restricting evaluation to English trained models. Although the CDP DATASET has disclosures in other languages it represents a small portion of the reports. We plan to include relevant climate change datasets from the multilingual European Union Public Data Catalog[9] in the future, while encouraging contributions from the broader community. Another limitation is that for our human evaluation pilot study we were able to only get results for a single model. We wish to build a small labeled dataset where climate experts map State climate action plans to their corresponding CDP questions for evaluation purposes. Doing such manual labeling is particularly difficult for CDP due to the large number of questions but this resource could then be used efficiently to evaluate multiple models and baselines.

We do not thoroughly investigate the efficiency-accuracy trade-offs of the Transformer models in this work. We provide the compute and training efficiency statistics in 6.8.2 and Table 6.10 as

[8]By construction, in our rating there may be multiple relevant questions found for each text segment.

[9]data.europa.edu

only a step in this direction. In this work we used the MiniLM model, a cross-encoder, for the CDP-QA experiments. Although this model is much smaller, at test time it requires a forward pass for each question-answer pair, which is computationally expensive. In future work it would be interesting to compare the cross-encoder to bi-encoders model architectures to better understand the accuracy vs. performance trade-off. We encourage future work on CLIMABENCH to leverage models that are both performant and efficient.

6.8 Appendix

6.8.1 Compute Details

We used a 24 core AMD Ryzen CPU machine with 128 GB RAM for data processing. For training and inference of the deep learning models, we utilize 4 Nvidia RTX 2080Ti GPUs with 11GB memory each. Each model was trained on a single GPU at a time.

Section	**Category/Label**
Hazards: Adaptation	Adaptation
Adaptation	Adaptation
Buildings	Buildings
Hazards: Climate Hazards	Climate Hazards
Hazards: Social Risks	Climate Hazards
Climate Hazards	Climate Hazards
Climate Hazards and Vulnerability	Climate Hazards
Climate Hazards & Vulnerability	Climate Hazards
City-wide Emissions	Emissions
Emissions Reduction	Emissions
GHG Emissions Data	Emissions
Local Government Emissions	Emissions
Emissions Reduction: City-wide	Emissions
City Wide Emissions	Emissions
Emissions Reduction: Local Government	Emissions
Local Government Operations GHG Emissions Data	Emissions
Energy Data	Energy
Energy	Energy
Food	Food
Governance and Data Management	Governance and Data Management
Opportunities	Opportunities
Strategy	Strategy
Urban Planning	Strategy
Transport	Transport
Waste	Waste
Water	Water
Water Security	Water

Table 6.7: The section topics in the CDP Cities Questionnaire and the corresponding Labels assigned by a climate expert.

Task	Average	Max	Min	Std
CLIMA-INS	203	4588	11	326
CLIMA-INS	206	4588	11	335
CLIMA-CDP	73	801	11	83
CLIMA-QA	105	834	15	88
CLIMATEXT	23	124	11	10
CLIMATESTANCE	30	98	11	12
CLIMATEENG	30	98	11	12
CLIMATEFEVER	47	311	11	19
SCIDCC	580	2014	13	223

Table 6.8: Statistics for the number of tokens in each task of CLIMABENCH

Model	Source	# Params	Vocab Size	Max Length
BERT	[35]	110M	30K	512
RoBERTa	[71]	125M	50K	512
DistilRoBERTa	[102]	82M	50K	512
Longformer	[13]	149M	50K	4096
SciBERT	[12]	110M	30K	512
ClimateBERT	[135]	82M	50K	512

Table 6.9: Pretrained Transformer Language Models used for Classification tasks

6.8.2 CO2 Emission Related to Experiments

A cumulative of 338 hours of computation was performed on hardware of type RTX 2080 Ti (TDP of 250W). Total emissions are estimated to be 36.5 kgCO_2eq. Estimations were conducted using the MachineLearning Impact calculator presented in [58].

Model	Avg. Runtime (in hours)	Avg. Train Samples/Second	Avg. Train Steps/Second
ClimateBERT	0.40	104.83	1.64
DistilRoBERTa	0.40	101.04	1.58
SciBERT	0.70	53.86	0.84
RoBERTa	0.80	50.46	0.79
BERT	0.85	49.32	0.77
Longformer	14.95	13.82	0.76

Table 6.10: Compute Efficiency Metrics for the Pretrained Transformer models for the experiments conducted on CLIMABENCH. Models based on the DistilRoBERTa architecture are the most efficient due to smaller model size.

6.8.3 Pretrained Transformer Models

BERT [35] is a popular Transformer-based language model pre-trained on masked language modeling and next sentence prediction tasks. It makes use of WordPiece tokenization algorithm that breaks a word into several subwords, such that commonly seen subwords can also be represented by the model.
RoBERTa [71] uses dynamic masking and eliminates the next sentence prediction pre-training task, while using a larger vocabulary and pre-training on much larger corpora compared to BERT. Another notable difference is the use of byte pair encoding compared to wordPiece in BERT.
DistilRoBERTa [102] leverages knowledge distillation during the pre-training phase reducing the size of the RoBERTa model by 40%, while retaining 97% of its language understanding capabilities and being 60% faster. Sanh et al. [102] originally distilled the BERT model but we utilize the better performing RoBERTa version in our experiments.
Longformer [13] extends Transformer-based models to support longer sequences with the help of sparse-attention. It uses a combination of local attention and global attention mechanism that allows for linear attention complexity and thus makes it feasible to run on longer documents (max 4096 tokens). It however takes much longer to train than the shorter context (512 tokens) models.
SciBERT [12], a pretrained language model based on BERT, leverages unsupervised pretraining on a large multi-domain corpus of scientific publications to improve performance on downstream scientific NLP tasks. It was evaluated on tasks like sequence tagging, sentence classification and dependency parsing with datasets from scientific domains. SciBERT gives significant improvements over BERT on these datasets.
ClimateBERT [135] was warm-started from the DistilRoBERTa model and pretrained on text corpora from climate-related research paper abstracts, corporate and general news and reports from companies that were not publicly released with the model. It was evaluated on tasks like sentiment analysis (using a private dataset), and public datasets like CLIMATEXT and CLIMATEFEVER. In this paper, we evaluate and compare the performance of ClimateBERT on diverse CC tasks for the first time, providing a comprehensive, publicly available and reproducible evaluation.

Parameter	Values
loss	hinge, squared_hinge
C	0.01, 0.1, 1, 10
class_weight	none, balanced

Table 6.11: For the linear SVM, we grid search over the parameters with 5-fold validation to get the best fit out of 80 candidates (16 values * 5 folds) with F1 Macro as the scoring mechanism

6.9 Climate Text Sources

The reports considered here include climate assessments, climate legislation, agency reports, regulatory filings, climate action plans (CAPs), and corporate ESG (Environmental, Social, and Governance) and CSR (Corporate Social Responsibility) documents [1, 24].

	CDP-CITIES		CDP-STATES		CDP-CORP	
Model	**MRR@10**	**MRR@All**	**MRR@10**	**MRR@All**	**MRR@10**	**MRR@All**
No Finetuning on CDP						
BM25	0.055	0.077	0.084	0.105	0.153	0.180
MiniLM	0.099	0.118	0.120	0.142	0.320	0.342
Finetuned on CDP						
	In-Domain		In-Domain		In-Domain	
ClimateBERT	0.331	0.344	0.422	0.431	0.753	0.754
MiniLM	**0.366**	**0.378**	0.482	0.491	**0.755**	**0.757**
Best Model Finetuned on all						
MiniLM	0.352	0.364	**0.489**	**0.497**	0.745	0.747

Table 6.12: MRR@k scores for BM25, ClimateBERT and MSMARCO-MiniLM on the three subsets of CLIMA-QA. Models finetuned and evaluated on same subset fall under In-Domain.

	CDP-STATES		CDP-CORP	
Model	**MRR@10**	**MRR@All**	**MRR@10**	**MRR@All**
No Finetuning				
BM25	0.084	0.105	0.153	0.180
MiniLM	0.120	0.142	0.320	0.342
Finetuned on CDP-CITIES				
	Transfer		Transfer	
ClimateBERT	0.298	0.314	0.465	0.477
MiniLM	**0.353**	**0.366**	**0.489**	**0.500**

Table 6.13: MRR@k scores for BM25, ClimateBERT and MSMARCO-MiniLM on the Transfer experiments. Models are finetuned on CDP-CITIES and evaluated on States and Corporations.

A key step in curbing emissions and mitigating climate change has been the development of standards and frameworks for climate reporting such as GRI (Global Reporting Initiative), TCFD (Task Force on Climate-Related Financial Disclosures), CDP (Carbon Disclosure Project), SASB (Sustainability Accounting Standards Board), and SDG (Sustainable Development Goals) [2, 16, 20, 100]

For example, the World Resources Institute has built Climate Watch [3] to keep track of progress and commitments nations have made under the 2015 Paris Agreement. One element of their work has been the manual labeling of Nationally Determined Contributions (NDCs) with a number of descriptors including cross references to the UN Sustainable Development Goals which strongly overlap with the categories in our CDP dataset and task.

Question	MRR@132
Please provide details of your climate actions in the Agriculture sector.	0.870
Please provide details of your climate actions in the Waste sector.	0.789
Please provide details of your climate actions in the Transport sector.	0.774
Please provide details of your climate actions in the Buildings & Lighting sector.	0.597
Please describe these current and/or anticipated impacts of climate change.	0.492
Please complete the table below.	0.487
Please indicate the opportunities and describe how the region is positioning itself to take advantage of them.	0.445
Please provide details of your climate actions in the Energy sector.	0.397
Please describe the adaptation actions you are taking to reduce the vulnerability of your region's citizens, businesses and infrastructure to the impacts of climate change identified in 6.6a.	0.378
Please describe these current and/or future risks due to climate change.	0.327
List any emission reduction, adaptation, water related or resilience projects that you have planned within your region for which you hope to attract financing, and provide details on the estimated costs and status of the project. If your region does not have any relevant projects, please select "No relevant projects" under Project Area.	0.319
Please provide details of your climate actions in the Land use sector.	0.286
Please provide the details of your region-wide base year emissions reduction target(s). You may add rows to provide the details of your sector-specific targets by selecting the relevant sector in the sector field.	0.252
Please describe the adaptation actions you are taking to reduce the vulnerability of your region's citizens, businesses and infrastructure to the risks due to climate change identified in 5.4a.	0.247

Table 6.14: Question difficulty evaluated on the test set of CDP-STATES ranked from best performing to worst performing. Filtered to only questions that appeared at least twenty times.

Although SDGs were first established by the United Nations to measure the progress of nation states towards development goals, they have been adopted by both corporations and regional and local jurisdictions to measure their sustainability efforts.

However, since the cross-referencing with SDGs is largely voluntary many cities, for example, have CAPs that are hundreds of pages in length but that provide no alignment with SDGs. Being able to effectively align the text between different climate documents to the various standards and disclosure frameworks is a critical component of climate policy evaluation and a real-world challenge. See also Stede and Patz [111] for more in-depth information.

Ex.	**Text Segment from State Climate Action Plans**	**Top Questions from CDP-STATES (using fine-tuned MiniLM)**
1	Sea level rise will inundate some nearby coastal areas, and related salt-water intrusion, coupled with increased drought stress may impact water supplies.	Q1: Please describe the current and/or anticipated impacts of climate change. Q3: Please detail any compounding factors that may worsen the impacts of climate change in your region.
2	The afforestation goal is to increase the area of forested lands in the state by 50,000 acres annually through 2025.	Q1: Please provide the details of your region's target(s). Q2: Please provide details of your climate actions in the Land use sector.
3	State law defines environmental justice as the fair treatment of people of all races, cultures, and incomes with respect to the development, adoption, implementation, and enforcement of environmental laws, regulations, and policies.	Q1: Please explain why you do not have policies on deforestation and/or forest degradation. Q4: Please provide details of your climate actions in the Governance sector.
4	By a majority vote, the ICCAC presents a policy option that, if deemed necessary, would build one new 1200-megawatt nuclear power plant in Iowa by January 1, 2020.	Q3: Please provide details of your renewable energy or electricity target(s). Q4: Please provide details of your climate actions in the Energy sector.
5	California maintains a GHG inventory that is consistent with IPCC practices ... Reports from facilities and entities that emit more than 25,000 MTCO2e are verified by a CARB-accredited third-party verification body.	Q1: Please give the name of the primary protocol, standard, or methodology you have used to calculate your government's GHG emissions. Q3: Please provide the following information about the emissions verification process.
6	A leading driver of these high emissions is the fact that the District's daytime population swells by 400,000 workers every workday, which is the largest percentage increase in daytime population of any large city in the nation.	Q4: Please indicate if your region-wide emissions have increased, decreased, or stayed the same since your last emissions inventory, and please describe why. Q5: Please report your region-wide base year emissions in the table below.

Table 6.15: Examples from our human pilot study in which our climate expert has evaluated the relevance of CDP questions linked to selected text from state climate action plans. A fragment of the matched text is presented with two illustrative questions from the set of five question matches generated by our model. The first two examples show high degrees of success. In example 1, our model correctly identifies the state CAP text as impact-related and captures the specific discussion of compound risks. However, example 3 appears to highlight a gap in the CDP questionnaire related to the topic of environmental justice, a result in itself of considerable interest.

Chapter 7

Aligning Unstructured Paris Reports with SDG Framework

- Daniel Spokoyny and Jannele Cai and Tom Corringham and Taylor Berg-Kirkpatrick, "Aligning Unstructured Paris Reports with SDG Framework."

7.1 Introduction

In the final chapter we re-examine the Nationally Determined Contributions (NDC) reports from Chapter 5 in the modern era of Large Language Models. We saw in the previous Chapter that there are existing semi-structured climate questionnaires (Carbon Disclosure Project) that can be utilized as a source of weak supervision for finetuning models for information extraction tasks over climate documents such as aligning passages according to their CDP questions. However, as we have alluded earlier, the vast majority of climate documents are largely unstructured and non-standardized, yet are critically important for climate scientists, policymakers, and various stakeholders.

Specifically, the United Nations Paris Agreement of 2015 established a set of over 160 country specific Nationally Determined Contributions (NDCs), updated in 2021, and underscored an international commitment towards a more sustainable and resilient future. Both sets of NDCs laid out ambitious targets, reflecting a collective will to act against climate change. These *unstructured texts*, while critically important, vary widely in format and are dense with diplomatic and technical jargon, presenting an opportunity for computational methods to distill and map their content to tangible goals.

The United Nations Sustainable Development Goals (SDGs) offer a structured framework, comprising 17 goals and 169 sub targets that promote global well-being and environmental protection. Specifically, we consider the SDGs as a taxonomy with a hierarchical set of classes with natural language descriptions of each class, but it does not contain any explicit labeled data. Linking the commitments made in the NDCs to the SDGs allows for a coherent understanding of global sustainability aspirations and establishes a concrete pathway for tracking progress and implementation. Aligning unstructured climate documents to new taxonomies with limited labeled

data is a challenging task which appears frequently in the climate domain. In this work we seek to study methods for addressing this challenge.

First, can we simply use the same cross-encoder based models from the previous chapter to align the NDCs to the SDGs? In a small human evaluation we saw potential in using the cross-encoder to align unstructured state climate action plans to the CDP framework. However, this task now involves both unstructured documents and a new framework which the models have not seen before. We set out to empirically analyze this and find that although they generalize to some extent, they are a ways below human performance on this task.

Second, we seek to explore whether a new technology: Large Language Models (LLMs) offer a potential opportunity to address this challenge. Might they through few-shot prompting be able to classify sentences in the NDCs with their corresponding SDG goals and targets? Prompting LLMs provides a relatively unsophisticated and yet extremely powerful way to leverage the models' capabilities. Furthermore, today some of the most advanced LLMs are easily accessible through APIs and web interfaces, making them a potentially well-suited tool for a wide range of climate policy researchers.

As part of this study, we are going to utilize the World Resources Institute's Climate Watch dataset [85] which contains manual annotations from NDCs reports according to the SDG framework. Although this dataset is a valuable resource, through our analysis we will show that it only covers a small portion of the NDCs, motivating the need for NLP methods that could scale to the entire corpus. To achieve this, we will use our own annotated data that will 1) help validate and better understand the Climate Watch dataset 2) provide an evaluation benchmark where we can compare our models along with a measure of inter-annotator agreement.

Our contributions are as follows: 1) We conduct an empirical study of LLMs as well as cross-encoder architectures on the task of aligning NDCs to SDGs. 2) We introduce a benchmark for cross-comparing our models, annotators, and the existing Climate Watch dataset. 3) We analyze specific methods to further boost performance on this task.

Finally, we aim to release the entirety of the NDC reports along with their predicted SDG alignment as an artifact for the community to use. By doing so, we aim to bridge a gap between global commitments and tangible outcomes, to foster transparency and ensure that the aims of these international agreements are better understood, monitored, and ultimately realized.

7.2 Datasets

In this section we will introduce the World Resources Institute's Climate Watch dataset that we utilized for our experiments [85]. The dataset includes sentences from NDCs submitted before 2021, each of which are labeled with goals and targets. There are 17 goals, and 169 targets, each of which are associated with a goal. Statistics on the dataset are shown in Table 7.1 and Table 7.2.

Each sentence in the document is labeled with one of the 17 SDGs and one of the 169 targets. Some sentences may also be labeled with multiple goals or targets. Example sentences and their labels are shown in Table 7.3. In Figure 7.1 we show the distribution of *SDG-Goals* in the *Climate-Watch* dataset.

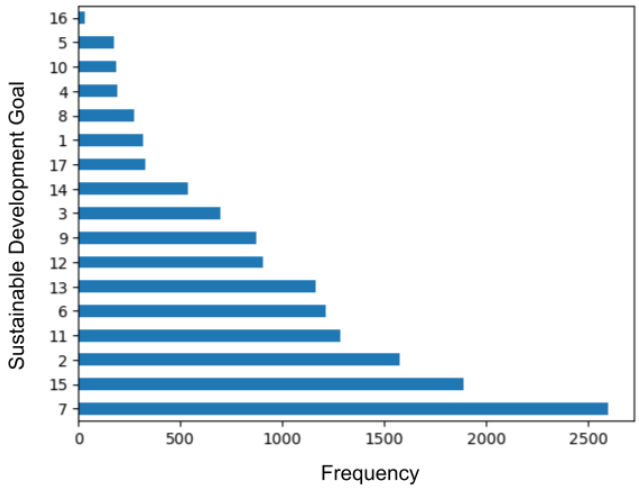


Figure 7.1: Histogram of the number of labels for each SDG in the Climate Watch dataset.

Table 7.1: Statistics for the Climate Watch dataset.

Property	Number
NDC Documents	214
Countries with Documents	186
Labelled Sentences	6813
Sentences with Multiple Goal Labels	1386
Sentences with Multiple Target Labels	2302

Preprocessing The *Climate-Watch* dataset has the SDG annotations, various associated metadata, and the raw text snippet from the NDC documents. However, these snippets are not directly linked to the exact locations in the NDC documents. We obtain a dataset of the full texts of the NDC documents as HTML files and using simple heuristics were able to match 94.8% of the annotations to their exact document spans. In Figure 7.2 we plot the distribution of where in the NDC documents the *Climate-Watch* annotations are found.

Table 7.2: Statistics for sentences in the Climate Watch dataset.

Property	Mean
Sentence Length (characters)	137.2
Labelled Sentences per Document	66.4
Goals per Sentence	1.34
Targets per Sentence	1.49

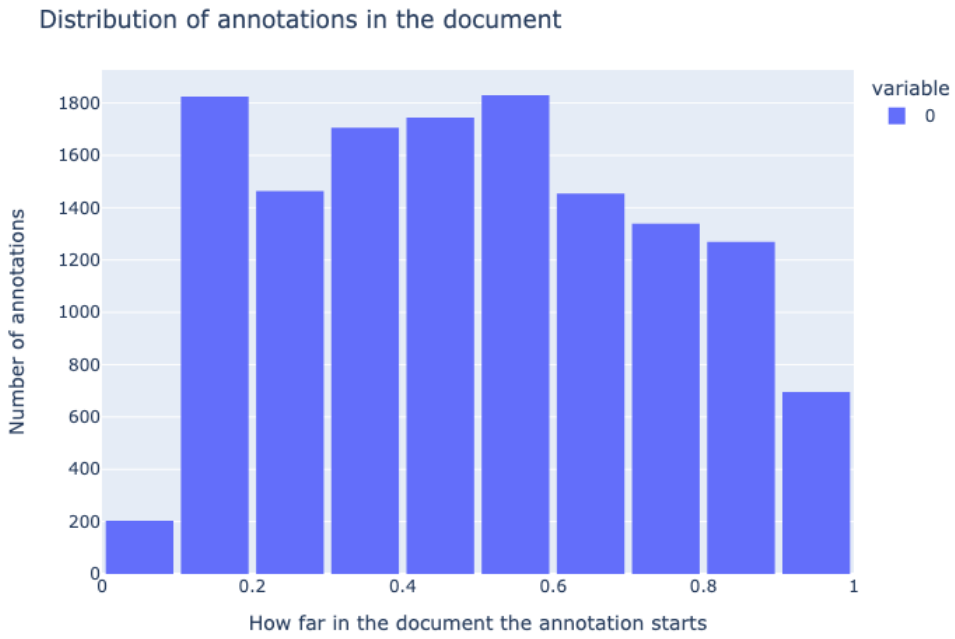


Figure 7.2: Histogram of where in the NDC documents the *Climate-Watch* annotations are found.

Table 7.3: *SDG-Goal* and *SDG-Target* labels of example sentences from the *Climate-Watch* dataset.

Climate Watch Labelled Examples	Goals	Targets
Reduce rural peoples' dependence on fuel for cooking and heating.	12	12.2
Reduce fuel consumption through efficiency standards	7, 11	7.3, 11.2
Guyana will implement other policies to encourage energy efficiency and the use of renewable energy, including building codes and net-metering of residential renewable power.	7	7.2, 7.3

7.2.1 Constructing Additional Benchmarks

We also created two small evaluation datasets that we will use to benchmark various aspects of our prompting strategies.

To construct the *Data-Random* dataset, we preprocess the HTML version of the NDC reports, using the NLTK sentencizer on the the HTML tags that contain the majority of the textual content (<p> and <li>). We further filter the sentences to be between 80 and 300 characters in length. Across all of the reports, this yields over 100,000 sentences. From this set, we randomly sampled 120 sentences to be labeled by our annotators.

To construct the *Data-Balanced* dataset, we selected 5 random annotations from *Climate-Watch* for each of the 17 *SDG-Goals* . The *Data-Random* and the *Data-Balanced* were drawn from 32 and 53 NDC reports, respectively.

Both of these datasets were subsequently labeled by three separate manual annotators, comprising one expert climate scientist and two university students with some climate policy understanding. Each sentence was independently labeled with up to three *SDG-Goals* that the annotator believed were most relevant to the sentence. For the *Data-Random* dataset, annotators could optionally select a "not relevant" label if they believed the sentence did not align with any of the *SDG-Goals* .

Inter Annotator Agreement

Later, in Section 7.3.1 we will use the *Data-Random* to estimate the portion of the NDC documents that have been labeled in the *Climate-Watch* dataset. Whereas, the *Data-Balanced* dataset will allow us to compare the performance of both our models and annotators against a balanced set of the *Climate-Watch* dataset.

Using our annotators, we show in Figure 7.3 the distribution of the predicted *SDG-Goals* for the *Data-Random* dataset, which we can contrast with the distribution of *SDG-Goals* in the *Climate-Watch* dataset (Figure 7.2). We found the most common *SDG-Goals* in *Data-Random* were 13, 15, 7 whereas in the *Climate-Watch* dataset it is 7, 15 and 2. The SDG Goal 13 (Take urgent action to combat climate change and its impacts) could be interpreted very broadly and thus our annotators ended up selecting it for a variety of sentences.

For the *Data-Random* split we calculated the inter-annotator agreement using Cohen's kappa (which has a range of -1 to 1) between the expert and each of the novices as (0.629, 0.524) [29]. However, on the *Data-Balanced* the agreement was lower (K = 0.215, K = 0.179), reflecting disparate annotation strategies among the annotators. Notably, some annotators demonstrated a conservative approach, opting to select only the primary goal, whereas others exhibited more leniency in their selections.

7.3 Experiments

In this section, we introduce our experiments in which we use different prompting strategies with GPT models to classify sentences according to SDG . We will use ChatGPT-3.5 and GPT-4-Turbo as our main models to conduct prompt-based classification experiments. We will use JSON-mode API option to ensure the model outputs are properly structured for classification tasks. As our

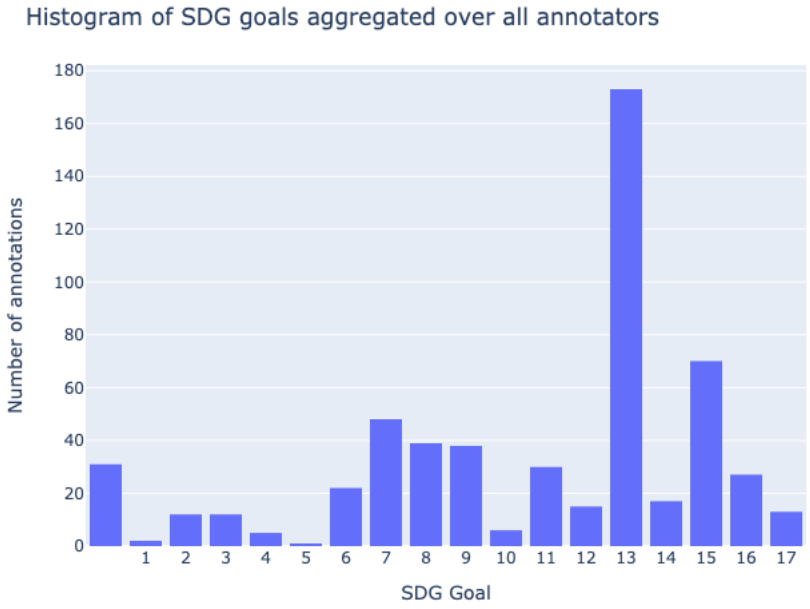


Figure 7.3: Histogram of the predicted *SDG-Goals* for the *Data-Random* dataset aggregated across all annotators.

zero-shot classification baselines we will use MiniCDP , the cross-encoder model finetuned on the semi-structured CDP questionnaire data from Chapter 6 as well as its base model architecture MiniLM model.

7.3.1 *Data-Random*

First, using our manual annotations we will try to estimate the existing coverage of the *Climate-Watch* dataset. We found that out of 120 sentences, 13 were labeled non-relevant by the Expert and 8 were labeled as not-relevant by at least two of the annotators. Since there are on average 724 sentences per document and only 66.4 sentences are labeled in the *Climate-Watch* dataset, we estimate that only 10-15% of the NDC have been labeled. We show a histogram of the predicted *SDG-Goals* for the *Data-Random* dataset in Figure 7.3.

Although this is a very rough estimate, it clearly shows that the vast majority remains unlabeled and motivates the need for a more scalable approach to labeling these documents. Although, to our knowledge, there is no full description of the methodology used to construct the *Climate-Watch* dataset, Northrop et al. [85] suggests that keyword searches were used to select climate actions.

Following this analysis, we aim to also measure how LLMs perform compared to our annotators on this random subset of sentences from the NDC documents. To do so we construct a simple prompt to predict a single *SDG-Goal* for each sentence. We have a simple instruction:

```
Given the following Input Text predict the Sustainable Development
    Goal (label) out of the following 17 options:
```

Table 7.4: Results on single *SDG-Goal* prediction for the *Data-Random* dataset.

Annotator	Accuracy
Annotator-1	80.0%
Annotator-2	70.8%
Model	Avg
ChatGPT-3.5	72.5%
GPT-4-Turbo	75.0%

Table 7.5: Results on multiple *SDG-Goal* prediction for the *Data-Random* dataset.

Annotator	Jaccard
Annotator-1	0.59
Annotator-2	0.50
Model	Avg
ChatGPT-3.5	0.55
GPT-4-Turbo	0.59

followed by listing out all of the *SDG-Goals* (see Table 7.12). To further encourage the model to produce well-formatted JSON outputs, we include an output specification in the prompt:

```
Generate a json object like so: {\'label\': [\'2\']}
```

And lastly, to capture non-relevant sentences, we include "0: None of the above labels are applicable" as an option in the list of *SDG-Goals* as well.

As models we use ChatGPT-3.5 and GPT-4-Turbo with the same prompt. We find that GPT-4-Turbo predicted 6 out of 13 non-relevant sentences correctly, while ChatGPT-3.5 was unable to predict any of them. Upon closer inspection, we found that ChatGPT-3.5 predicted a very general goal, (Goal-13: Take urgent action to combat climate change and its impacts), for a majority of non-relevant sentences.

To evaluate the performance of the models we calculate the accuracy as whether the model's prediction matched one of the Expert labels. We show results in Table 7.4. We find that both models perform well with GPT-4-Turbo being slightly better. We also include the other two annotators as a point of reference although it is not a direct comparison, as annotators were allowed to select up to three *SDG-Goals* .

For a more fair comparison, we simply modify the prompt output specification

```
Generate a json object like so: {\'label\': [\'1\', \'2\']}
```

to allow the models to predict multiple *SDG-Goals* . We use the Jaccard similarity to measure the overlap between the sets of *SDG-Goals* . We show the results in Table 7.5. From the results, we see that on random sentences from the NDC documents, both GPT models perform at similar levels to the annotators.

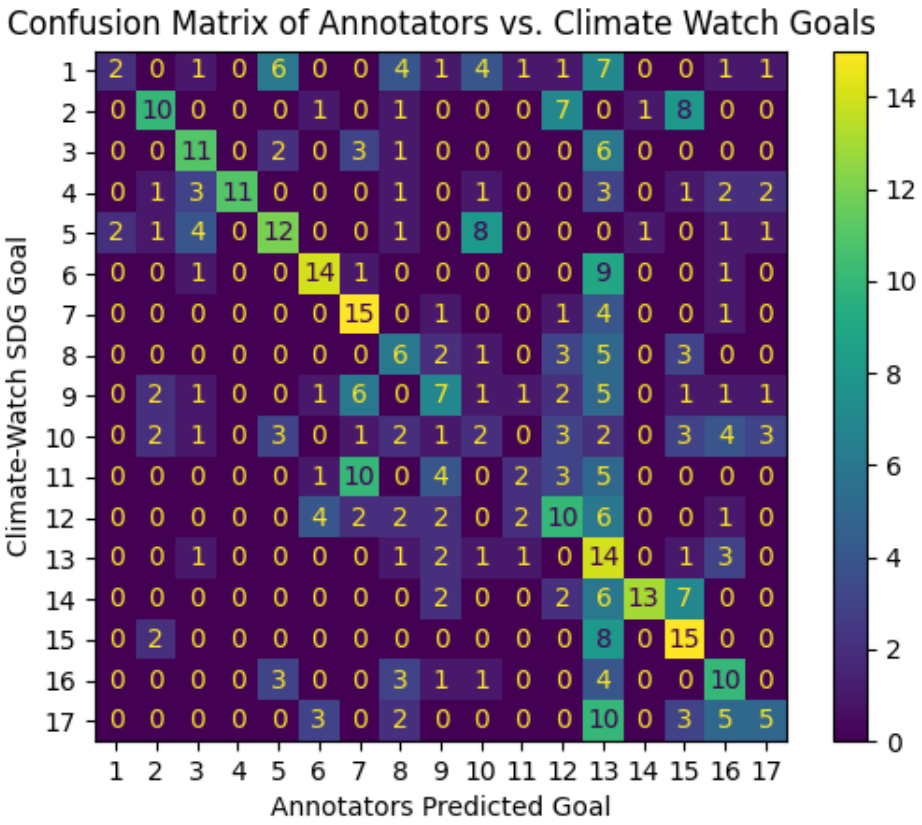


Figure 7.4: Confusion matrix for the *Data-Balanced* dataset.

7.3.2 *Data-Balanced*

First we want to compare the performance of our annotators against the annotations from the *Climate-Watch* dataset. As our metric, we report whether the percentage of sentences where annotators selected the same *SDG-Goal* as the *Climate-Watch* dataset. For our three annotators we found this to be 49.4%, 57.6%, and 48.2%. By using a balanced dataset, we can also evaluate the average accuracy of our annotators for each *SDG-Goal* shown in Table 7.6 along with a confusion matrix in Figure 7.4.

In Table 7.7 we compare the performance of our models on the *Data-Balanced* dataset. We find that with the top scoring *SDG-Goal* the MiniCDP model achieves an accuracy of 30.6% while the MiniLM model is almost 9% lower at 21.1%. Both of the LLMs perform much better with the ChatGPT-3.5 model achieving 47.1% and the GPT-4-Turbo model achieving 49.4%.

Since in the *Data-Balanced* split there is only a single *SDG-Goal* label for each sentence, we also aim to quantify how well the models perform against our annotators with multiple *SDG-Goal* label predictions. For the MiniLM and MiniCDP models, we simply take the models' top three scoring goal predictions.

We select the annotator with the highest accuracy against the *Climate-Watch* labels to compare our model predictions against. We use the Jaccard similarity to measure the overlap between the

Table 7.6: Average Annotator Performance by *SDG-Goal* on the *Data-Balanced* dataset.

SDG-Goal	Avg
7	93.33
15	86.67
6	86.67
13	80.00
5	73.33
3	66.67
2	66.67
4	66.67
16	53.33
14	46.67
12	40.00
8	33.33
9	26.67
17	26.67
1	13.33
11	13.33
10	6.67

sets of *SDG-Goals* . The results are presented in Table 7.8.

We again find that the MiniCDP model to be slightly better than the MiniLM model with Jaccard scores of 0.19 and 0.17, respectively. While both of the other annotators have Jaccard scores of 0.46, the GPT models achieve higher similarity scores of 0.48 and 0.50.

7.3.3 *Climate-Watch*

Although, the *Data-Random* and *Data-Balanced* data splits are relatively small, we have found that prompting GPT models to predict *SDG-Goals* is a promising approach for classifying sentences. In our final set of experiments, we will use the *Climate-Watch* dataset to benchmark prediction of *SDG-Targets* . From the full *Climate-Watch* dataset we randomly selected 200 sentences and in this section will refer to it as the ground truth.

We explore two modes for predicting the *SDG-Targets* , *oracle*: where we use the ground truth *SDG-Goal* label to sub-select only the corresponding *SDG-Targets* , and *full*: where we predict all *SDG-Targets* for a given sentence. We prompt the models to produce the *SDG-Target* labels as JSON objects. Since many sentences have multiple *SDG-Target* labels, for our metric we use the Jaccard similarity. Results for these experiments are shown in Table 7.9.

For the *full* mode, we see that GPT-4-Turbo is substantially better than ChatGPT-3.5 with Jaccard scores of 0.42 and 0.28, respectively. As expected, in the *oracle* mode both models perform better with the gap between the two models slightly decreasing.

Table 7.7: Single *SDG-Goal* prediction results for the *Data-Balanced* dataset.

Annotator	Avg
Expert	49.4%
Annotator-1	57.6%
Annotator-2	48.2%
Model	
MiniLM	21.1%
MiniCDP	30.6%
ChatGPT-3.5	47.1%
GPT-4-Turbo	49.4%

Table 7.8: Multi *SDG-Goals* prediction results for the *Data-Balanced* dataset compared to top performing annotator.

Annotator	Jaccard
Annotator-1	0.46
Annotator-2	0.46
Model	
MiniLM	0.17
MiniCDP	0.19
ChatGPT-3.5	0.48
GPT-4-Turbo	0.50

Table 7.9: Multi *SDG-Targets* prediction results for the Climate Watch dataset.

Model	Jaccard
ChatGPT-3.5 *full*	0.28
GPT-4-Turbo *full*	0.42
ChatGPT-3.5 *oracle*	0.49
GPT-4-Turbo *oracle*	0.57

Table 7.10: Multi *SDG-Target* prediction results with in-context learning for the *Climate-Watch* dataset.

Model	Number ICL	Jaccard
ChatGPT-3.5	1	0.31
ChatGPT-3.5	10	0.35
ChatGPT-3.5	20	0.36
GPT-4-Turbo	20	0.44

Table 7.11: Multi *SDG-Target* prediction results with expert prompting on the *Climate-Watch* dataset.

Model	Jaccard
ChatGPT-3.5 *full*	0.27
GPT-4-Turbo *full*	0.42
ChatGPT-3.5 *oracle*	0.52
GPT-4-Turbo *oracle*	0.58

In Context Learning

One of the most desirable features of modern LLMs is their ability to use task-specific examples in their prompt to further boost performance. In the next set of experiments, we additionally provide up to 20 in-context learning (ICL) examples to both of our models. We show the results in Table 7.10.

We find that the ChatGPT-3.5 model improves with additional ICL examples, getting much closer to the performance of the GPT-4-Turbo model. In contrast the 20 ICL examples only slightly improve the performance of the GPT-4-Turbo model.

Prompting Strategies

There are a variety of prompting techniques that have been shown improve performance such chain of thought [136], maieutic prompting [52], or self-ask [50]. Xu et al. [142] found that providing a model with a prompt that describes an identity of distinguished expert can improve performance. We experiment with a simple form of *expert-prompting* for a climate policy expert. We generated the expert identity using GPT-4 using an example from Xu et al. [142], and added "You are a climate policy expert..." to the beginning of our instruction. The results are shown in Table 7.11 and the full expert-prompt is shown in the Appendix. We find that there is a small improvement for both models in the *oracle* mode but no effect in the *full* mode.

7.4 Artifact

To enable climate researchers to we use the best existing system/configuration we identified to annotate the entire NDC documents according to the SDG Goals and Targets. We will aim to provide the annotations, in a structured format along with the original NDC documents.

7.5 Related Work

7.5.1 NDC SDG Linking

There is research that has explored connecting NDCs and SDGs but it has predominantly been through manual expert annotations Policymakers across several jurisdictions observe that there is significant overlap between the implementation process for SDGs and NDCs, and that the linking of both policymaking processes increases the efficacy of climate policy design. Northrop et al. [85] and Brandi et al. [19] provide detailed evidence for the convergence between SDGs and NDCs. Antwi-Agyei et al. [7] aim to leverage the alignments and misalignments between West African NDCs and global SDGs to increase the efficacy of West African climate policies.

However, due to the painstaking effort required to align these documents most studies are limited in scope: concentrating on a specific geographical region [7] or selecting a single or subset of the SDG goals [42, 105]. In contrast our study we have significant coverage across all: SDG Goals and Targets, geographical regions provided the availability of NDC document in English language, the entire texts of the documents. Additionally, some approaches utilize keyword search or extraction techniques to label data, however, these methods have limitations [54], including potential biases introduced by the choice of keywords.

7.5.2 NLP for Climate

Research in applying/building NLP tools for climate-related tasks has largely focused on peer-reviewed academic papers, climate finance documents [75], and non-climate texts such as Wikipedia. Most recently, Smith et al. [105] analyzed peer-reviewed scientific articles published between 2001 and 2020 which are indexed as relevant to SDG 3 and one or more SDGs in Dimensions, "the most exhaustive database for scientific publications" (Smith et al., 2023). They used results from an existing machine learning method to classify scientific publications by their SDG relevance.

There has been exploratory work on using ChatGPT to interact with climate documents such as the Intergovernmental Panel on Climate Change Report (IPCC)[124]. In contrast, our aim is to understand how modern LLMs with zero shot prompting, or few-shot in-context-learning could assist in these tasks.

7.6 Conclusion

In summary we have constructed benchmarks which allowed us to compare the performance of models, annotators, using the *Climate-Watch* dataset on unstructured NDC documents. Using

this data we were able to show that existing manual efforts are low coverage and motivating the need for automated methods. We then found that our finetuned cross-encoder model from the previous Chapter was still slightly better than its underlying base model, although the improvement was marginal. Finally, we saw across various experiments that by prompting GPT models we were able to match the performance of our annotators on *SDG-Goal* and *SDG-Target* prediction. Overall, these findings highlight the potential of leveraging machine learning models, particularly GPT-based ones, to effectively annotate unstructured climate documents such as the NDCs. To enable climate researchers, we use the best existing configuration we identified to annotate the entire NDC documents according to the SDG Goals and Targets. We will aim to provide the annotations, in a structured format along with the original NDC documents.

Table 7.12: The 17 Sustainable Development Goals.

Goal	Description
1	End poverty in all its forms everywhere
2	End hunger, achieve food security and improved nutrition and promote sustainable agriculture
3	Ensure healthy lives and promote well-being for all at all ages
4	Ensure inclusive and equitable quality education and promote lifelong learning opportunities for all
5	Achieve gender equality and empower all women and girls
6	Ensure availability and sustainable management of water and sanitation for all
7	Ensure access to affordable, reliable, sustainable and modern energy for all
8	Promote sustained, inclusive and sustainable economic growth, full and productive employment and decent work for all
9	Build resilient infrastructure, promote inclusive and sustainable industrialization and foster innovation
10	Reduce inequality within and among countries
11	Make cities and human settlements inclusive, safe, resilient and sustainable
12	Ensure sustainable consumption and production patterns
13	Take urgent action to combat climate change and its impacts
14	Conserve and sustainably use the oceans, seas and marine resources for sustainable development
15	Protect, restore and promote sustainable use of terrestrial ecosystems, sustainably manage forests, combat desertification, and halt and reverse land degradation and halt biodiversity loss
16	Promote peaceful and inclusive societies for sustainable development, provide access to justice for all and build effective, accountable and inclusive institutions at all levels
17	Strengthen the means of implementation and revitalize the global partnership for sustainable development

7.7 Appendix

One Full Goal Prompt with 5 ICL examples.

```
Given the following Input Text predict the Sustainable
   Development Goal (goal) out of the following 17 options:
Sustainable Development Goal
1: End poverty in all its forms everywhere
2: End hunger, achieve food security and improved nutrition and
   promote sustainable agriculture
3: Ensure healthy lives and promote well-being for all at all
   ages
```

4: Ensure inclusive and equitable quality education and promote lifelong learning opportunities for all
5: Achieve gender equality and empower all women and girls
6: Ensure availability and sustainable management of water and sanitation for all
7: Ensure access to affordable, reliable, sustainable and modern energy for all
8: Promote sustained, inclusive and sustainable economic growth, full and productive employment and decent work for all
9: Build resilient infrastructure, promote inclusive and sustainable industrialization and foster innovation
10: Reduce inequality within and among countries
11: Make cities and human settlements inclusive, safe, resilient and sustainable
12: Ensure sustainable consumption and production patterns
13: Take urgent action to combat climate change and its impacts
14: Conserve and sustainably use the oceans, seas and marine resources for sustainable development
15: Protect, restore and promote sustainable use of terrestrial ecosystems, sustainably manage forests, combat desertification, and halt and reverse land degradation and halt biodiversity loss
16: Promote peaceful and inclusive societies for sustainable development, provide access to justice for all and build effective, accountable and inclusive institutions at all levels
17: Strengthen the means of implementation and revitalize the global partnership for sustainable development
Input Text: By 2026, a study will have been developed on the impacts derived from climate change on agricultural and fishing production systems, including effects on agricultural health, and whose results are shared appropriately to the realities and worldviews of the different communities.
goal:13
Input Text: establishment of information centers for farmers that provides guidance on adaptive management of agriculture; etc
goal:2
Input Text: These additional mitigation actions will be achieved through focusing on energy sector and industrial processes, as well as conservation and development of forests, sustainable agriculture and waste management. "Market-based mechanisms" and transfer of environment friendly technologies under the legal regime of UNFCCC as well as transfer of management practices, play a key role in successful and result oriented conditional mitigation actions.

```
goal:15
Input Text: reforestation and afforestation for the
   rehabilitation of degraded lands.
goal:15
Input Text: public awareness is being raised on the need for
   rationalizing water use
goal:12
Input Text: Save water for irrigation by using plastic films/
   mulches on potato and vegetable fields;
goal:
```

Below is the full expert prompt that we used in our experiments.

```
You are a climate policy expert specializing in understanding the
   complexities of climate systems and the impacts of human
   activities. Your knowledge spans climate science, mitigation, and
    adaptation strategies. You excel in analyzing research findings
   and developing policies that balance scientific evidence,
   political realities, and societal needs. Your expertise is
   instrumental in crafting effective and equitable climate policies
    at all levels, driving action towards a sustainable and
   resilient future.
```

Chapter 8

Conclusion

In this dissertation we have outlined the different challenges and opportunities in applying NLP to the climate-related domain.

8.1 Numeracy

Numbers appear in text in all sorts of contexts, and different parts of the NLP community have been working on different aspects of understanding numbers in text.

Strides have been made studying NLP for math word problems as test-bed of how well models can reason about latent structures in a controlled setting. In this sub-field, the numbers are usually treated as symbolic variables for manipulation. Separately, but along side this, a separate sub-field emerged of pushing the boundaries of what mathematical reasoning can be done with neural networks. From the other side of the spectrum, NLP works aimed at factual generation has examined better grounding of numerical facts in text. In this thesis we carved out a relatively unexplored and yet crucial aspect of understanding numbers in text: numerical commonsense.

Before, numerical commonsense was encompassed by the general notion of commonsense in NLP, with simple numerical facts such as knowing that a cat has four legs or that there are 12 months in a year. These types of facts can be trivially solved through either memorization or lookups to a knowledge base. However, in this thesis we explore quantities that are deeply embedded in documents, and in some domains (academic papers, news articles, etc.) constitute the vast majority of numerical mentions. When encountered with these quantities, they may be unique and appearing for the first time in the text. These quantities may have explicit types such as measurements with physical units, or they may have implicit types such as "cars produced at a facility".

To understand these quantities, we need to understand 1) the surrounding context with which they appear, 2) the types of quantities they represent, and 3) the relationships they have with other quantities in the text. The metrics we use to train and evaluate LLMs models were designed for language composed of solely discrete symbols or words. Yet this overlooks the simple fact that quantities are not discrete symbols, but are in fact continuous. The same way that using perplexity as a core metric has been instrumental for benchmarking LLMs and driving language model research, numerical commonsense needs its own set of tasks and metrics to drive progress in this

field. With this as context, we offer a different framing of our numeracy work: as contributing new tasks along with metrics that are more suited for evaluating numerical commonsense for NLP models.

In our research, we have reached several conclusions about the capabilities of deep learning models in learning and understanding numerical information. While these models have the potential to learn numerical commonsense, our findings show that they are not always effective. By building custom reasoning modules and carefully considering training objectives we have shown that it is possible to learn this type of knowledge. However, just as it does not make practical sense to learn conversion factors, we should aim to imbue as much of this type-level knowledge into the model as possible.

8.1.1 Numeracy and LLMs

Although, modern model architectures have had no specialized mechanisms for handling quantities, with possibly the exception of tokenizing numbers using digits, they have surpassed across a wide range of numerical reasoning tasks. However, it is unclear to what extent these models are learning numerical commonsense, and a full evaluation of this is beyond the scope of this thesis. It is conceivable that at a large enough scale these models capture symbolic reasoning, numerical commonsense, and are able to recall numerical facts as well. However, it could also be the case that by integrating continuous representations of quantities at the input and output level, the models fare even better while being more sample efficient. One promising avenue for future work worth exploring is post-hoc modifying these existing models to perform better on numerical commonsense tasks.

8.1.2 LLMs and Tool Use

One increasingly favorable aspect of modern LLMs is the integration of "Tools." These tools can take various forms, such as "schedule an appointment for Monday," "search Google for," and "send an email to Robert." Essentially, tools allow an LLM to initiate arbitrary computations through text output and receive the results in raw text form. This concept is incredibly powerful, facilitating the development of more advanced AI assistants and even more sophisticated theorem provers. In the context of numeracy, this capability has enabled models to perform complex numerical computations by invoking Python programs. Consequently, LLMs do not need to execute these mathematical operations themselves, as there are already simple, efficient, and accurate methods for such computations. In this thesis, we similarly refrained from having the model learn proper unit conversion calculations when studying measurements. The tasks we examined, such as masked measurement infilling, are not solvable using existing tools and thus provide a signal for learning better numerical representations. The true benefit of using LLMs with tools lies in the complementary strengths of these systems, which can combine different forms of fuzzy and precise reasoning effectively.

8.1.3 Numeracy and Retrieval Augmented Measurement Prediction

Whereas, in this thesis we have studied quantities within a short context window of a document they appear in, this presupposes that the quantities of interest are already present in the surrounding context. One of the most exciting future directions would be to break this restriction, and consider the case of retrieving information from a large corpus of text. Using a separate numerical retriever could also be viewed as a specialised tool to help improve numerical commonsense. We speculate that the task of masked quantity infilling as the task could provide strong enough signal to train a retriever to represent quantities in a way that is unique and distinct from retrievers built for general language tasks. These representations could then be helpful for a variety of downstream use cases, such as numerical question answering, fact checking, information retrieval, and more.

8.2 NLP for Climate Documents

In looking at applying NLP to the climate domain, we have examined different strategies to bolster NLP models with domain-specific weakly supervision: using document structure, semi-structured questionnaires, and few-shot prompting. Climate domain experts require the whole gamut of workflows and tools ranging from exhaustively labeling documents to quickly searching across large collections for specific relevant information. The same strategies we used to improve cross-encoders for the climate domain, by leveraging weak-supervision in structured surveys, could also enhance existing LLMs as well. In the last chapter, we showed that by prompting LLMs one could match human annotator performance on a real world climate text classification task. And advances on better text representation using LLMs such as LLM2Vec [10] could allow for increased performance on large scale corpus data mining.

With the all of the benefits of LLMs, there arise plenty of challenges in successfully adopting them in the climate domain. These billion parameter models are out of reach for most organizations to deploy. Many of the best performing models are proprietary, and the closed-source nature of these models restricts and hampers certain types of research. By not knowing the training data, it could lead to misinterpretation of generalization vs. memorization. The competitive landscape further fuels restrictions on the use of these models, such as not being able to record true output probabilities, or lack of full control surrounding decoding strategies. On the other hand, open-source models are steadily improving, and many organizations devote resources to make inference with these models as accessible as with the proprietary models.

Prompting of LLMs is also a double-edged sword. There is an abundance of research on designing prompting strategies, selecting in-context learning examples, or using LLMs as judges. However, each of these carries potential pitfalls and biases that are not yet fully understood. And unlike with past NLP technologies, the failures of LLMs are silent and not easily detectable. What can not be overstated however, is that due to the natural language interface it provides, prompting could be utilized by climate researchers without requiring them to be experts in NLP. However, these by no means are these challenges insurmountable and should not deter the community from adapting these models to the climate domain. One exciting area of future work lies in HCI studies that can help communities of climate experts build high-quality datasets by interacting with LLMs.

8.2.1 Broader Impacts and Proliferation

The largest headway for impact in the climate domain lay not in the models themselves, but in the development of cleanly labeled datasets, well-formulated tasks, and the development of tools that can support climate domain experts without requiring them to be experts in NLP. This is by no means a problem unique to the climate domain, in fact, Thakur et al. [114] showed the problems that arise due to annotation selection bias in the information retrieval community. Evaluating against these benchmarks artificially underestimated the performance of deep learning models. In the climate domain, the use of keyword-based methods for labeling data is still prevalent.

Finally, the most important aspect of this work is identifying individuals or organizations capable of serving as a bridge between these two communities. This requires having strong domain expertise in a climate-related domain, and an overall understanding of data science, machine learning, and NLP. Organizations such as the Climate Policy Radar or Climate Change AI through various initiatives are able to scale these sorts of interdisciplinary, multi-stakeholder projects. Getting involved early in the task formulation and annotation process, is crucial for future work in this domain and maximizing potential impact.

Bibliography

[1] 2004. Who cares wins: Connecting the financial markets to a changing world? Technical report, United Nations, The Global Compact. 6.9

[2] 2022. 2022 status report. Technical report, TCFD. 6.9

[3] 2022. Climate watch. Technical report, World Resources Institute, Washington, D.C. 5.2, 6.9

[4] Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. MathQA: Towards interpretable math word problem solving with operation-based formalisms. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2357–2367, Minneapolis, Minnesota. Association for Computational Linguistics. 4.4.1

[5] Daniel Andor, Luheng He, Kenton Lee, and Emily Pitler. 2019. Giving bert a calculator: Finding operations and arguments with reading comprehension. In *EMNLP/IJCNLP*. 2.5

[6] Daniel Ansari. 2016. Number symbols in the brain. In *Development of Mathematical Cognition*, pages 27–50. Elsevier. 2.2.1

[7] Philip Antwi-Agyei, Andrew J. Dougill, Thomas P. Agyekum, and Lindsay C. Stringer. 2018. Alignment between nationally determined contributions and the sustainable development goals for West Africa. *Climate Policy*, 18(10):1296–1312. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/14693062.2018.1431199. 7.5.1

[8] Tarik Arici, Kushal Kumar, Hayreddin Çeker, K K Saladi, and Ismail B. Tutar. 2021. Solving price per unit problem around the world: Formulating fact extraction as question answering. In *KDD TrueFact Workshop*. 3.1

[9] Nicolas Baya-Laffite and Jean-Philippe Cointet. 2016. Mapping Topics in International Climate Negotiations: A Computer-Assisted Semantic Network Approach. In Sebastian Kubitschko and Anne Kaun, editors, *Innovative Methods in Media and Communication Research*, pages 273–291. Springer International Publishing, Cham. 5.1

[10] Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. Llm2vec: Large language models are secretly powerful text encoders. 8.2

[11] Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A Pretrained Language Model for Scientific Text. *arXiv:1903.10676 [cs]*. ArXiv: 1903.10676. 5.3

[12] Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics. 6.4, **??**, 6.8.3

[13] Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *ArXiv*, abs/2004.05150. 6.4, **??**, 6.8.3

[14] Lea Berrang-Ford, Anne J Sietsma, Max W. Callaghan, Jan C. Minx, Pauline F. D. Scheelbeek, Neal Robert Haddaway, Andy Haines, and Alan D. Dangour. 2021. Systematic mapping of global research on climate and health: a machine learning review. *The Lancet. Planetary Health*, 5:e514 – e525. 6.2

[15] Robbert Biesbroek, Shashi Badloe, and Ioannis N. Athanasiadis. 2020. Machine learning for research on climate change adaptation policy integration: an exploratory UK case study. *Regional Environmental Change*, 20(3):85. 5.1

[16] Amy Bills, Beth Mackay, Chang Deng-Beck, George Bush, Maia Kutner, Rachel Carless, and Simeran Bachra. 2022. Protecting people and the planet: Putting people at the heart of city climate action. Technical report, CDP. 6.9

[17] Julia Anna Bingler, Mathias Kraus, and Markus Leippold. 2021. Cheap Talk and Cherry-Picking: What ClimateBert has to say on Corporate Climate Risk Disclosures. *SSRN Electronic Journal*. 5.1

[18] Julia Anna Bingler, Mathias Kraus, and Markus Leippold. 2021. Cheap talk and cherry-picking: What climatebert has to say on corporate climate risk disclosures. *Corporate Finance: Governance*. 6.1

[19] Clara Brandi, Adis Dzebo, and Hannah Janetschek. 2017. The case for connecting the implementation of the paris climate agreement and the 2030 agenda for sustainable development. 7.5.1

[20] Halina Szejnwald Brown, Martin de Jong, and Teodorina Lessidrenska. 2007. The rise of the global reporting initiative (gri) as a case of institutional entrepreneurship. Working Paper 36, John F. Kennedy School of Government, Harvard University. 6.9

[21] C2ES. 2020. U.S. State Climate Action Plans, Center for Climate and Energy Solutions. 5.6

[22] C40. 2021. C40 Climate Action Planning Resource Centre. 5.6

[23] Daniel Fernando Campos, Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, Li Deng, and Bhaskar Mitra. 2016. Ms marco: A human generated machine reading comprehension dataset. *ArXiv*, abs/1611.09268. 6.4

[24] Archie B. Carroll. 2009. A history of corporate social responsibility: Concepts and practices. In Andrew Crane, Dirk Matten, Abagail McWilliams, Jeremy Moon, and Donald S. Siegel, editors, *The Oxford Handbook of Corporate Social Responsibility*. Oxford University Press, Oxford. 6.9

[25] Flávio N Cação, Anna Helena Reali Costa, Natalie Unterstell, Liuca Yonaha, Taciana Stec,

and Fábio Ishisaki. 2021. Deeppolicytracker: Tracking changes in environmental policy in the brazilian federal official gazette with deep learning. In *ICML 2021 Workshop on Tackling Climate Change with Machine Learning*. 6.2

[26] Arun Chaganty and Percy Liang. 2016. How much is 131 million dollars? putting numbers in perspective with compositional descriptions. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 578–587, Berlin, Germany. Association for Computational Linguistics. 3.5.1

[27] Ilias Chalkidis, Abhik Jana, Dirk Hartung, Michael James Bommarito, Ion Androutsopoulos, Daniel Martin Katz, and Nikolaos Aletras. 2022. Lexglue: A benchmark dataset for legal language understanding in english. In *ACL*. 6.2

[28] Chung-Chi Chen, Hen-Hsen Huang, Hiroya Takamura, and Hsin-Hsi Chen. 2019. Numeracy-600K: Learning numeracy for detecting exaggerated information in market comments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6307–6313, Florence, Italy. Association for Computational Linguistics. 2.1, 2.2.2, 2.2.3, 2.5, 3.5.1, 4.4.2

[29] Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20:37 – 46. 7.2.1

[30] Tom Corringham, Daniel Spokoyny, Eric Xiao, Christopher Cha, Colin Lemarchand, Mandeep Syal, Ethan Olson, and Alexander Gershunov. 2021. Bert classification of paris agreement climate action plans. In *ICML 2021 Workshop on Tackling Climate Change with Machine Learning*. 6.2

[31] Dominique M David-Chavez and Michael C Gavin. 2018. A global assessment of Indigenous community engagement in climate research. *Environmental Research Letters*, 13(12):123005. 5.5

[32] Stanislas Dehaene, Ghislaine Dehaene-Lambertz, and Laurent Cohen. 1998. Abstract representations of numbers in the animal and human brain. *Trends in Neurosciences*, 21:355–361. 2.2.1

[33] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics. 2.1, 2.2.2, 2.4.3

[34] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]*. ArXiv: 1810.04805. 5.1

[35] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics. 6.4, **??**,

6.8.3

[36] Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proc. of NAACL*. 2.5

[37] Abhijeet Dubey, Lakshya Kumar, Arpan Somani, Aditya Joshi, and Pushpak Bhattacharyya. 2019. "when numbers matter!": Detecting sarcasm in numerical portions of text. In *Proceedings of the Tenth Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 72–80, Minneapolis, USA. Association for Computational Linguistics. 4.4.2

[38] Yanai Elazar and Yoav Goldberg. 2019. Where's my head? Definition, data set, and models for numeric fused-head identification and resolution. *Transactions of the Association for Computational Linguistics*, 7:519–535. 4.4.2

[39] Yanai Elazar and Yoav Goldberg. 2019. Where's my head? definition, data set, and models for numeric fused-head identification and resolution. *Transactions of the Association for Computational Linguistics*, 7:519–535. 3.5.1

[40] Yanai Elazar, A. Mahabal, Deepak Ramachandran, Tania Bedrax-Weiss, and Dan Roth. 2019. How large are lions? inducing distributions over quantitative attributes. In *ACL*. 2, 3.5.1

[41] Noriko. Fujiwara, Harro van Asselt, Stefan Böβner, Sebastian Voigt, Niki-Artemis Spyridaki, Alexandros Flamos, Emilie Alberola, Keith Williges, Andreas Türk, and Michael ten Donkelaar. 2019. The practice of climate change policy evaluations in the european union and its member states: results from a meta-analysis. *Sustainable Earth*, 2:1–16. 6.2

[42] Natalya D. Gallo, David G. Victor, and Lisa A. Levin. 2017. Ocean commitments under the Paris Agreement. *Nature Climate Change*, 7(11):833–838. 5.5, 7.5.1

[43] Mor Geva, Ankit Gupta, and Jonathan Berant. 2020. Injecting numerical reasoning skills into language models. In *ACL*. 2.5, 3.5.1

[44] Mor Geva, Ankit Gupta, and Jonathan Berant. 2020. Injecting numerical reasoning skills into language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 946–958, Online. Association for Computational Linguistics. 4.3

[45] Justin Grimmer and Brandon M. Stewart. 2013. Text as Data: The Promise and Pitfalls of Automatic Content Analysis Methods for Political Texts. *Political Analysis*, 21(3):267–297. 5.1

[46] Yuxian Gu, Robert Tinn, Hao Cheng, Michael R. Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2022. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3:1 – 23. 6.2

[47] Vinh Thinh Ho, Yusra Ibrahim, Koninika Pal, Klaus Berberich, and Gerhard Weikum. 2019. Qsearch: Answering quantity queries from text. In *SEMWEB*. 3.1

[48] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy:

Industrial-strength Natural Language Processing in Python. 5.2

[49] Danqing Huang, Shuming Shi, Chin-Yew Lin, Jian Yin, and Wei-Ying Ma. 2016. How well do computers solve math word problems? large-scale dataset construction and evaluation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 887–896, Berlin, Germany. Association for Computational Linguistics. 4.4.1

[50] Jiaxin Huang, Shixiang Shane Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2022. Large language models can self-improve. *ArXiv*, abs/2210.11610. 7.3.3

[51] Zhihua Jin, Xin Jiang, Xingbo Wang, Qun Liu, Yong Wang, Xiaozhe Ren, and Huamin Qu. 2021. Numgpt: Improving numeracy ability of generative pre-trained models. *ArXiv*, abs/2109.03137. 3.5.1, 3.5.1

[52] Jaehun Jung, Lianhui Qin, Sean Welleck, Faeze Brahman, Chandra Bhagavatula, Ronan Le Bras, and Yejin Choi. 2022. Maieutic prompting: Logically consistent reasoning with recursive explanations. *ArXiv*, abs/2205.11822. 7.3.3

[53] A. Kalyan, Abhinav Kumar, Arjun Chandrasekaran, Ashish Sabharwal, and Peter Clark. 2021. How much coffee was consumed during emnlp 2019? fermi problems: A new reasoning challenge for ai. *ArXiv*, abs/2110.14207. 3.5.1

[54] Gary King, Patrick Lam, and Margaret E. Roberts. 2017. Computer-Assisted Keyword and Document Set Discovery from Unstructured Text. *American Journal of Political Science*, 61(4):971–988. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/ajps.12291. 7.5.1

[55] Andrei P. Kirilenko and Svetlana O. Stepchenkova. 2014. Public microblogging on climate change: One year of twitter worldwide. *Global Environmental Change*, 26:171–182. 6.2

[56] Julian Kölbel, Markus Leippold, Jordy Rillaerts, and Qian Wang. 2020. Does the CDS Market Reflect Regulatory Climate Risk Disclosures? *SSRN Electronic Journal*. 5.1, 5.5

[57] Julian F. Kölbel, Markus Leippold, Jordy Rillaerts, and Qian Wang. 2020. Does the cds market reflect regulatory climate risk disclosures. 6.2

[58] Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. 2019. Quantifying the carbon emissions of machine learning. *arXiv preprint arXiv:1910.09700*. 6.8.2

[59] Guillaume Lample and François Charton. 2020. Deep learning for symbolic mathematics. In *International Conference on Learning Representations*. 2.5

[60] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. *ArXiv*, abs/1909.11942. 2.1, 2.2.2

[61] Dennis Lee, Christian Szegedy, Markus Rabe, Sarah Loos, and Kshitij Bansal. 2020. Mathematical reasoning in latent space. In *International Conference on Learning Representations*. 2.5

[62] Jane A Leggett. 2020. The United Nations Framework Convention on Climate Change, the Kyoto Protocol, and the Paris Agreement: A Summary. Technical Report R46204, Congressional Research Service. 5.1

[63] Markus Leippold and Thomas Diggelmann. 2020. Climate-fever: A dataset for verification of real-world climate claims. In *NeurIPS 2020 Workshop on Tackling Climate Change with Machine Learning*. 6.1, 6.2, **??**, 6.3.3

[64] Markus Leippold and Francesco Saverio Varini. 2020. Climatext: A dataset for climate change topic detection. In *NeurIPS 2020 Workshop on Tackling Climate Change with Machine Learning*. 6.1, 6.2, **??**, 6.3.3

[65] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics. 4.3

[66] Bill Yuchen Lin, Seyeon Lee, Rahul Khanna, and Xiang Ren. 2020. Birds have four legs?! numersense: Probing numerical commonsense knowledge of pre-trained language models. *ArXiv*, abs/2005.00683. 3.5.1, 4.1, 4.4.2

[67] Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. *arXiv preprint arXiv:1705.04146*. 2.5

[68] Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 158–167, Vancouver, Canada. Association for Computational Linguistics. 4.4.1

[69] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*. 2.2.2, 3.2.5

[70] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692. 4.3

[71] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692. 6.4, **??**, 6.8.3

[72] Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel S. Weld. 2020. S2ORC: The Semantic Scholar Open Research Corpus. In *Proceedings of ACL*. 2.3

[73] Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *ICLR*. 3.4, 4.3

[74] Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *ICLR*. 6.5.1

[75] Alexandra Luccioni, Emily Baylor, and Nicolas Duchene. 2020. Analyzing Sustainability Reports Using Natural Language Processing. ArXiv:2011.08073 [cs]. 7.5.2

[76] Alexandra Luccioni and Hector Palacios. 2019. Using Natural Language Processing to

Analyze Financial Climate Disclosures. 5.1, 5.5

[77] Alexandra Sasha Luccioni, Emily Baylor, and Nicolas Anton Duchêne. 2020. Analyzing sustainability reports using natural language processing. *ArXiv*, abs/2011.08073. 6.1, 6.2

[78] Andreas Madsen and Alexander Rosenberg Johansen. 2020. Neural arithmetic units. In *International Conference on Learning Representations*. 2.5

[79] Paul Meddeb, Stefan Ruseti, Mihai Dascalu, Simina Terian, and Sébastien Travadel. 2022. Counteracting french fake news on climate change using language models. *Sustainability*. 6.1

[80] Paramita Mirza, Simon Razniewski, Fariz Darari, and Gerhard Weikum. 2017. Cardinal virtues: Extracting relation cardinalities from text. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 347–351, Vancouver, Canada. Association for Computational Linguistics. 2.1

[81] Prakamya Mishra and Rohan Mittal. 2021. Neuralnere: Neural named entity relationship extraction for end-to-end climate change knowledge graph construction. In *ICML 2021 Workshop on Tackling Climate Change with Machine Learning*. **??**, 6.3.3

[82] Swaroop Mishra, Arindam Mitra, Neeraj Varshney, Bhavdeep Sachdeva, Peter Clark, Chitta Baral, and Ashwin Kalyan. 2022. NumGLUE: A suite of fundamental yet challenging mathematical reasoning tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3505–3523, Dublin, Ireland. Association for Computational Linguistics. 4.1

[83] Nafise Sadat Moosavi, Andreas Rücklé, Dan Roth, and Iryna Gurevych. 2021. Learning to reason for text generation from scientific tables. *arXiv preprint arXiv:2104.08296*. 3.1

[84] Aakanksha Naik, Abhilasha Ravichander, Carolyn Rose, and Eduard Hovy. 2019. Exploring numeracy in word embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3374–3380, Florence, Italy. Association for Computational Linguistics. 2.5, 3.5.1

[85] Eliza Northrop, Hana Biru, Sylvia Lima, Mathilde Bouyé, and Ranping Song. 2016. Examining the Alignment between the Intended Nationally Determined Contributions and Sustainable Development Goals. 7.1, 7.2, 7.3.1, 7.5.1

[86] Kuntal Kumar Pal and Chitta Baral. 2021. Investigating numeracy learning ability of a text-to-text transfer model. *ArXiv*, abs/2109.04672. 3.5.1

[87] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830. 6.5.1

[88] Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, Eliza Rutherford, Tom Hennigan, Jacob Menick, Albin Cassirer, Richard Powell, George van den Driessche, Lisa Anne Hendricks, Maribeth Rauh, Po-Sen Huang, Amelia Glaese, Johannes Welbl, Sumanth Dathathri, Saffron Huang, Jonathan Uesato, John F. J. Mellor, Irina Higgins,

Antonia Creswell, Nathan McAleese, Amy Wu, Erich Elsen, Siddhant M. Jayakumar, Elena Buchatskaya, David Budden, Esme Sutherland, Karen Simonyan, Michela Paganini, L. Sifre, Lena Martens, Xiang Lorraine Li, Adhiguna Kuncoro, Aida Nematzadeh, Elena Gribovskaya, Domenic Donato, Angeliki Lazaridou, Arthur Mensch, Jean-Baptiste Lespiau, Maria Tsimpoukelli, N. K. Grigorev, Doug Fritz, Thibault Sottiaux, Mantas Pajarskas, Tobias Pohlen, Zhitao Gong, Daniel Toyama, Cyprien de Masson d'Autume, Yujia Li, Tayfun Terzi, Vladimir Mikulik, Igor Babuschkin, Aidan Clark, Diego de Las Casas, Aurelia Guy, Chris Jones, James Bradbury, Matthew G. Johnson, Blake A. Hechtman, Laura Weidinger, Iason Gabriel, William S. Isaac, Edward Lockhart, Simon Osindero, Laura Rimell, Chris Dyer, Oriol Vinyals, Kareem W. Ayoub, Jeff Stanway, L. L. Bennett, Demis Hassabis, Koray Kavukcuoglu, and Geoffrey Irving. 2021. Scaling language models: Methods, analysis & insights from training gopher. *ArXiv*, abs/2112.11446. 4.1

[89] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv e-prints*. 2.1, 2.2.2

[90] Abhilasha Ravichander, Aakanksha Naik, Carolyn Rose, and Eduard Hovy. 2019. Equate: A benchmark evaluation framework for quantitative reasoning in natural language inference. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 349–361. 3.5.1

[91] Abhilasha Ravichander, Aakanksha Naik, Carolyn Rose, and Eduard Hovy. 2019. EQUATE: A benchmark evaluation framework for quantitative reasoning in natural language inference. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 349–361, Hong Kong, China. Association for Computational Linguistics. 4.4.2

[92] Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. 6.4

[93] Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389. 6.5.3

[94] Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A Primer in BERTology: What we know about how BERT works. *arXiv:2002.12327 [cs]*. ArXiv: 2002.12327. 5.1

[95] David Rolnick, Priya L. Donti, Lynn H. Kaack, Kelly Kochanski, Alexandre Lacoste, Kris Sankaran, Andrew Slavin Ross, Nikola Milojevic-Dupont, Natasha Jaques, Anna Waldman-Brown, Alexandra Luccioni, Tegan Maharaj, Evan D. Sherwin, S. Karthik Mukkavilli, Konrad P. Kording, Carla Gomes, Andrew Y. Ng, Demis Hassabis, John C. Platt, Felix Creutzig, Jennifer Chayes, and Yoshua Bengio. 2019. Tackling Climate Change with Machine Learning. *arXiv:1906.05433 [cs, stat]*. ArXiv: 1906.05433. 5.1

[96] David Rolnick, Priya L. Donti, Lynn H. Kaack, Kelly Kochanski, Alexandre Lacoste, Kris Sankaran, Andrew Slavin Ross, Nikola Milojevic-Dupont, Natasha Jaques, Anna Waldman-Brown, Alexandra Sasha Luccioni, Tegan Maharaj, Evan D. Sherwin, S. Karthik Mukkavilli, Konrad P. Kording, Carla P. Gomes, Andrew Y. Ng, Demis Hassabis, John C.

Platt, Felix Creutzig, Jennifer Chayes, and Yoshua Bengio. 2022. Tackling climate change with machine learning. *ACM Comput. Surv.*, 55(2). 6.2

[97] Subhro Roy and Dan Roth. 2016. Solving general arithmetic word problems. *arXiv preprint arXiv:1608.01413*. 2.5

[98] Subhro Roy, Tim Vieira, and Dan Roth. 2015. Reasoning about quantities in natural language. *Transactions of the Association for Computational Linguistics*, 3:1–13. 4.4.2

[99] Pablo Ruiz, Clément Plancq, and Thierry Poibeau. 2016. Climate Negotiation Analysis. Technical Report hal-01423299, Jagiellonian University and Pedagogical University, Cracovie, Poland. 5.1

[100] Jeffrey D. Sachs. 2012. From millennium development goals to sustainable development goals. *The Lancet*, 379(9832):2206–2211. 6.9

[101] Swarnadeep Saha, Harinder Pal, and Mausam. 2017. Bootstrapping for numerical open IE. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 317–323, Vancouver, Canada. Association for Computational Linguistics. 2.1

[102] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108. 6.4, **??**, 6.8.3

[103] Sunita Sarawagi and Soumen Chakrabarti. 2014. Open-domain quantity queries on web tables: annotation, response, and consensus models. *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 3.1

[104] David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. 2019. Analysing mathematical reasoning abilities of neural models. In *International Conference on Learning Representations*. 2.2.1, 2.2.3, 2.5

[105] Thomas Bryan Smith, Raffaele Vacca, Luca Mantegazza, and Ilaria Capua. 2023. Discovering new pathways toward integration between health and sustainable development goals with natural language processing and network science. *Globalization and Health*, 19(1):44. 7.5.1, 7.5.2

[106] Georgios Spithourakis and Sebastian Riedel. 2018. Numeracy for language models: Evaluating and improving their ability to predict numbers. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2104–2115, Melbourne, Australia. Association for Computational Linguistics. 4.4.2

[107] Georgios P Spithourakis and Sebastian Riedel. 2018. Numeracy for language models: Evaluating and improving their ability to predict numbers. *arXiv preprint arXiv:1805.08154*. 2.1, 2.2.3, 2.5, 3.5.1

[108] Daniel Spokoyny and Taylor Berg-Kirkpatrick. 2020. An empirical investigation of contextualized number prediction. In *EMNLP*. 3.2.5, 3.5.1, 4.4.2

[109] Daniel Spokoyny, Ivan Lee, Zhao Jin, and Taylor Berg-Kirkpatrick. 2022. Masked measurement prediction: Learning to jointly predict quantities and units from textual context. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 17–29,

Seattle, United States. Association for Computational Linguistics. 4.3, 4.3.2

[110] Aarohi Srivastava and et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *ArXiv*, abs/2206.04615. 4.4.1

[111] Manfred Stede and Ronny Patz. 2021. The climate change debate and natural language processing. In *Proceedings of the 1st Workshop on NLP for Positive Impact*, pages 8–18, Online. Association for Computational Linguistics. 6.9

[112] Dhanasekar Sundararaman, Vivek Subramanian, Guoyin Wang, Liyan Xu, and Lawrence Carin. 2022. Improving downstream task performance by treating numbers as entities. 4.4.2

[113] Pradip Swarnakar and Ashutosh Modi. 2021. Nlp for climate policy: Creating a knowledge platform for holistic and effective climate action. *ArXiv*, abs/2105.05621. 6.2

[114] Nandan Thakur, Nils Reimers, Andreas Ruckl'e, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *ArXiv*, abs/2104.08663. 6.4, 8.2.1

[115] Avijit Thawani, Jay Pujara, and Filip Ilievski. 2021. Numeracy enhances the literacy of language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6960–6967, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics. 3.1, 3.2.5, 3.3, 3.5.1, 3.8.1, 3.8.5, 4.2.2

[116] Avijit Thawani, Jay Pujara, Filip Ilievski, and Pedro Szekely. 2021. Representing numbers in NLP: a survey and a vision. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–656, Online. Association for Computational Linguistics. 3.5.1

[117] Harold Thimbleby and Paul Cairns. 2010. Reducing number entry errors: solving a widespread, serious problem. *Journal of the Royal Society Interface*, 7(51):1429–1439. 2.1, 2.4.2

[118] James Thorne and Andreas Vlachos. 2017. An extensible framework for verification of numerical claims. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 37–40, Valencia, Spain. Association for Computational Linguistics. 2.1

[119] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *NAACL-HLT*. 6.3.3

[120] Andrew Trask, Felix Hill, Scott E Reed, Jack Rae, Chris Dyer, and Phil Blunsom. 2018. Neural arithmetic logic units. In *Advances in Neural Information Processing Systems*, pages 8035–8044. 2.5

[121] Walter R. Tribett, Ross J. Salawitch, Austin P. Hope, Timothy P. Canty, and Brian F. Bennett. 2017. Paris INDCs. In *Paris Climate Agreement: Beacon of Hope*, pages 115–146. Springer International Publishing, Cham. Series Title: Springer Climate. 5.1

[122] UNFCCC. 2016. Synthesis report on the aggregate effect of the intended nationally determined contributions. Technical Report FCCC/CP/2016/2, UNFCCC. 5.1

[123] UNFCCC. 2021. NDC Registry (interim). 5.2, 5.6

[124] Saeid Ashraf Vaghefi, Qian Wang, Veruska Muccione, Jingwei Ni, Mathias Kraus, Julia Anna Bingler, Tobias Schimanski, Chiara Colesanti-Senni, Nicolas Webersinke, Christrian Huggel, and Markus Leippold. 2023. chatclimate: Grounding conversational ai in climate science. *ArXiv*, abs/2304.05510. 7.5.2

[125] Roopal Vaid, Kartikey Pant, and Manish Shrivastava. 2022. Towards fine-grained classification of climate change related social media text. In *ACL*. 6.1

[126] Roopal Vaid, Kartikey Pant, and Manish Shrivastava. 2022. Towards fine-grained classification of climate change related social media text. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 434–443, Dublin, Ireland. Association for Computational Linguistics. **??**, **??**, 6.3.3

[127] Francesco S. Varini, Jordan Boyd-Graber, Massimiliano Ciaramita, and Markus Leippold. 2021. ClimaText: A Dataset for Climate Change Topic Detection. *arXiv:2012.00483 [cs]*. ArXiv: 2012.00483. 5.1

[128] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc. 6.4

[129] Tommaso Venturini, Nicolas Baya Laffite, Jean-Philippe Cointet, Ian Gray, Vinciane Zabban, and Kari De Pryck. 2014. Three maps and three misunderstandings: A digital mapping of climate diplomacy. *Big Data & Society*, 1(2):205395171454380. 5.1

[130] Eric Wallace, Yizhong Wang, Sujian Li, Sameer Singh, and Matt Gardner. 2019. Do nlp models know numbers? probing numeracy in embeddings. In *Empirical Methods in Natural Language Processing*. 2.5, 3.5.1

[131] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc. 6.2

[132] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics. 6.2

[133] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. 6.4

[134] Yan Wang, Xiaojiang Liu, and Shuming Shi. 2017. Deep neural solver for math word problems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 845–854, Copenhagen, Denmark. Association for Computational Linguistics. 4.4.1

[135] Nicolas Webersinke, Mathias Kraus, Julia Anna Bingler, and Markus Leippold. 2021. Climatebert: A pretrained language model for climate-related text. *ArXiv*, abs/2110.12010.

6.1, 6.4, **??**, 6.8.3

[136] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Huai hsin Chi, F. Xia, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903. 7.3.3

[137] John Whalen, Charles R Gallistel, and Rochel Gelman. 1999. Nonverbal counting in humans: The psychophysics of number representation. *Psychological science*, 10(2):130–137. 2.2.1

[138] Sarah Wiseman, Paul Cairns, and Anna Cox. 2011. A taxonomy of number entry error. In *Proceedings of HCI 2011 The 25th BCS Conference on Human Computer Interaction 25*, pages 187–196. 2.4.2

[139] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771. 2.2.2

[140] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics. 3.2.5

[141] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics. 6.5.1

[142] Benfeng Xu, An Yang, Junyang Lin, Quang Wang, Chang Zhou, Yongdong Zhang, and Zhendong Mao. 2023. Expertprompting: Instructing large language models to be distinguished experts. *ArXiv*, abs/2305.14688. 7.3.3

[143] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*. 2.2.2

[144] Dongxiang Zhang, Lei Wang, Luming Zhang, Bing Tian Dai, and Heng Tao Shen. 2019. The gap of semantic parsing: A survey on automatic math word problem solvers. *IEEE transactions on pattern analysis and machine intelligence*. 2.5

[145] Xikun Zhang, Deepak Ramachandran, Ian Tenney, Yanai Elazar, and Dan Roth. 2020. Do language embeddings capture scales? In *FINDINGS*. 3.2.5, 3.5.1, 3.5.1