Adversarially Robust Natural Language Inference using GPT-2

732A92 - Text Mining

Shashi Nagarajan (shana299)

August 28, 2022

Abstract

Resourced primarily by private sector concerns, large language models (LLMs) have been at the de facto centre of contemporary discourse on Natural Language Understanding for several years now. A number of these models are being deployed in real-life textual applications despite several reasonable questions as to their suitability for these applications, applications in which humans perform well naturally. At the heart of this issue is the fact that large models are ostensibly 'overfitted' to training data. Such models perform well on data that very closely resemble training data but poorly on data that for models may be 'adversarial' but none too different from training data for humans. In this project, I aim to make the LLM GPT-2 adversarially robust in the Natural Language Inference task by attempting to remove spurious artefacts (biases) in the Stanford Natural Language Inference (SNLI) dataset, through an algorithm called AFLite (which stands for Lightweight Adversarial Filtering). Because of resource constraints (compute and time), however, I do not succeed entirely. I do, however, make publicly available a hitherto unavailable repository of code that can further the objective, and point to some recent research for additional directions through which to progress towards adversarially robust NLI.

1. Introduction

The demonstration of the efficacy of transformer-based neural models (Vaswani et al. 2017) ushered in a new era of dominance of large language models (LLMs) in NLP. Ever since, (a handful of mostly private sector) institutions have been expending an enormous amount of resources (Bender et al. 2021) on developing LLMs, typically using massive datasets. This line of research remains motivated by dramatic performance gains LLMs have (reportedly) had on benchmark NLP tasks (A. Wang et al. 2018, A. Wang et al. 2019 and several others), after they are sensitised to specific tasks through fine-tuning and/or few-shot learning procedures.

On the other hand, a smaller community of researchers demonstrate that merely replacing some words chosen at random within input text sequences with synonyms, whilst largely preserving grammaticality, adversely impacts the performance of fine-tuned BERT (Devlin et al. 2019) models on NLP tasks, although human performance remained robust (Jin et al. 2020, Zang et al. 2020, Pruthi et al. 2019). It may indeed be well known that the issue of adversarial robustness of machine learning models is a broad concern (Szegedy et al. 2014, Goodfellow et al. 2015).

Nevertheless, the fact that LLM-based NLP is thriving (see table 1), despite the cited research and exorbitant pretraining costs, the issue of robust NLI particularly germane. A case may thus possibly be made for more stringent evaluation of the performance, particularly of fine-tuned LLMs than what is done in practice, which is the comparison of fine-tuned LLM performance versus the 'state-of-the-art' in benchmark tasks, without controlling for the existence of spurious artefacts in data (Le Bras et al. 2020, Niven et al. 2019).

Le Bras et al. 2020 apply AFLite (Sakaguchi et al. 2021), an algorithm to identify and filter out 'representation biases' in the context of Natural Language Inference (NLI, Bowman et al. 2015) through the fine-tuning of an LLM, namely RoBERTa-large (Y. Liu et al. 2019). One of their many demonstrations is that classifiers learned by fine-tuning RoBERTa-large on the AFLite filtered SNLI datasets (Bowman et al. 2015) yield poorer generalisation within in-distribution (SNLI) test datasets but better generalisation in out-of-distribution datasets including HANS (McCoy et al. 2019), NLI Diagnostics (included in GLUE, A. Wang et al. 2018), Stress tests (Naik et al. 2018) and Adversarial NLI (Nie et al. 2020) datasets, when compared to classifiers learned by fine-tuning RoBERTa-large on the full, unfiltered SNLI dataset as well as a random subset of the same.

These results establish the idea that the SNLI dataset contains representation biases with respect to RoBERTa-large, which help it perform well on the in-distribution (SNLI test) dataset and also that AFLite mitigates some of these biases, resulting in better generalisation on the NLI task, as proven by better performance on the out-of-distribution test datasets, thus enabling adversarially robust NLI. In this project, I attempt to extend the work of Le Bras et al. 2020 and check if AFLite identifies biases in the SNLI dataset when using classifiers learned by fine-tuning other LLMs as well. If it does, it would be reasonable to think of the LLMs as being much more robust to adversarial datasets. Moreover, the above-mentioned case for more stringent evaluations of the performance of LLMs would materialise strongly. Such an evaluation could entail mitigating representation biases in benchmark training datasets before reporting performance on test datasets. Re-evaluating the state-of-the-art in this manner can present a fairer picture of the performance of language models on natural language understanding tasks (NLU) and progress the NLU discourse positively.

Due to resource limitations (compute and time), however, this project is limited to evaluating the performance of GPT-2 based NLI models on the above-mentioned in-distribution and out-of-distribution datasets, after fine-tuning them for the NLI task using AFLite filtered, unfiltered and randomly subsetted SNLI training datasets. AFLite takes a number of hyper-parameters as inputs, and as such, a single run of AFLite on the SNLI training dataset, with a single non-trivial set of hyper-parameters, for an LLM such as RoBERTa-large, involves a formidably large number of computations[1]. In this project, therefore, the said evaluations shall be based only on a single set of AFLite hyper-parameters.

[1] For perspective, see section §§6.2, which reports the time taken for such a run

In terms of how the report is organised, §2 describes in some detail the AFLite algorithm, the LLM, GPT-2, the transfer learning methods relevant to this project, and R_K (Gorodkin 2004), the K-class generalisation of the Matthew's Correlation Coefficient (MCC), an evaluation metric used in the project. §3 sets out in some detail the datasets employed for fine-tuning and evaluation procedures. §4 reports implementation methodology and nuances, including the choice of hyper-parameters and the Python libraries packages used in the project. §5 points out the tables in the report that present the results of the procedures carried out in this project and summarises the same briefly. §6 presents a discussion consisting of a review of the project results from three perspectives, whilst also highlighting some limitations of the project and related work.

Through this limited project, AFLite could not be said to have identified (nor removed) representation biases in the SNLI training dataset, with respect to GPT-2; I have failed to carry out adversarially robust NLI using GPT-2. I do, however, make publicly available a code repository[2] that can be used off-the-shelf to run AFLite extensively on the SNLI training dataset with different GPT-2 models and various hyper-parameter choices, and collect ensuing results[3]. Such work can take forward what I do accomplish in this project and go on to enable adversarially robust NLI using GPT-2 and also weigh on whether in fact there is a strong case for stringent evaluation of LLM performance, as motivated above.

2. Theory

2.1. AFLite

In Machine Learning, models which are fit on large datasets, especially those with large number of parameters, tend to overfit examples from the data-rich 'head' of the distribution and underfit examples from the 'tail'. AFLite, short for 'Lightweight Adversarial Filtering', is a greedy algorithm presented by Sakaguchi et al. 2021 and Le Bras et al. 2020, whose objective is to filter out biases, specifically representation biases[4], in the data-rich head whilst preserving inherent complexities in the tail.

Formulation Suppose Φ is some feature representation defined over a dataset $\mathscr{D} = (X, Y)$ and $\mathscr{M}$ is a family of classification models that can be trained on subsets $S \subset \mathscr{D}$. AFLite's objective is to minimise the representation bias of Φ in S *w.r.t.* $\mathscr{M}$, denoted $\mathscr{R}(\Phi,\ S,\ \mathscr{M})$. That is, given a hyperparameter n, which denotes the target dataset size, i.e., the minimum size of S, the objective of AFLite is to *ideally* find:

$$\underset{S \subset \mathscr{D},\ |S| \geq n}{\arg\min}\ \mathscr{R}(\Phi,\ S,\ \mathscr{M}) \tag{1}$$

Denote by T and $S \setminus T$ two random partitions of a given subset S of the dataset $\mathscr{D}$. Further, denote as $q : 2^S \mapsto [0,1]$ the probability distribution over subsets $T \subset S$[5]. Finally, denote as $M_T \in \mathscr{M}$ a linear classifier trained on $S \setminus T$ with accuracy $f_{M_T}(\Phi(X^T), Y^T)$ on the held-out set $T = (X^T, Y^T)$. The presenters of AFLite, Le Bras et al. 2020, define $\mathscr{R}(\Phi,\ S,\ \mathscr{M})$ as:

$$\mathscr{R}(\Phi,\ S,\ \mathscr{M}) \triangleq \mathbb{E}_{T \sim q}[f_{M_T}(\Phi(X^T), Y^T)] \tag{2}$$

They further present mathematical simplification and reasoning that imply that under certain uniformity assumptions on q, this equation can be simplified as follows[6]:

$$\begin{gathered}\mathscr{R}(\Phi,\ S,\ \mathscr{M}) = \sum_{i \in S} \tilde{p}(i),\ \text{where} \\ \tilde{p}(i) \triangleq \frac{1}{|S|}\mathbb{E}_{T \subset S,\ T \ni i}[f_{M_T}(\Phi(X^T), Y^T)], \\ \text{and } \tilde{p}(i) \text{ is interpreted as a measure of how} \\ \textit{predictable} \text{ the instance } i \text{ is}\end{gathered} \tag{3}$$

This simplification reduces $\mathscr{R}(\Phi,\ S,\ \mathscr{M})$ from being a sum (expectation) over exponential number of summands ($T \in 2^S$) in Eq. (2) to another over a linear number of summands ($i \in S$) in Eq. (3). Nevertheless, AFLite's objective, shown in Eq. (1) involves a search in an exponentially large space, viz. $S \in 2^{\mathscr{D}}$. To make this more tractable, AFLite takes up a *greedy* approach, as described in the pseudo-code typeset as follows[7].

2.2. GPT-2

GPT-2, the LLM presented by Radford et al. 2019 differed from older LLMs in that it produced hitherto 'promising, competitive, and state of the art results [sic] depending on the [benchmark downstream] task', under *zero-shot* settings. GPT-2 was pre-trained on the (unmasked) language modelling task with a 40GB text corpus, the hitherto largest dataset for LLM pre-training, which its presenters believed to be diverse as well. Four versions of the model were pre-trained (and subsequently released), each with either 117M (GPT2-small), 345M (GPT2-medium), 762M (GPT2-large) or 1.5B (GPT2-xl) parameters, and the presenters reported increasing performance gains as model size increased.

In practice, GPT-2 requires natural language sequences to be encoded (tokenised) in a prescribed way before they are passed to the LLM as input. GPT-2 recognises a total of 50,257 tokens, an expansion when compared to previous generation LLMs. By way of architecture, GPT-2 employs multi-layer transformer decoders[8] (Vaswani et al. 2017). A number of other nuances such as residual connections,

[2] `https://github.com/shashiniyer/adversarial_nli_gpt2`

[3] Neither Sakaguchi et al. 2021 nor Le Bras et al. 2020 publish their implementations of AFLite

[4] These representation biases do *not* refer to ones caused by inequitable representation of sociological groups or similar; they take on a specific definition presented in the subsequent paragraphs

[5] 2^A denotes the power-set of set A

[6] $A \ni B$ denotes the event that set A contains set B and $i \in A$, that instance $(x_i, y_i) \in A$

[7] $A \uplus B$ denotes multiset addition of B with A; for example, $\{1,2,3\} \uplus \{3\} = \{1,2,3,3\}$

[8] Transformer decoder covered in lecture 4 and subsequent interactive session, hence not describing it

Year	Model	Main Language	Affiliate	# Parameters	Dataset Size
2018	GPT-1 (Radford et al. 2018)	English	Open AI	117M	5GB
2019	GPT-2 (Radford et al. 2019)	English	Open AI	1.5B	40GB
2019	BERT (Large) (Devlin et al. 2019)	English	Google	340M	16GB
2019	Megatron-LM (Shoeybi et al. 2019)	English	NVIDIA	8.3B	174GB
2020	ERNIE-GEN (Large) (Xiao et al. 2020)	English	Baidu	340M	16GB
2020	GPT-3 (Brown et al. 2020)	English	Open AI	175B	570GB
2021	GPT-J-6B (B. Wang 2021)	English	EleutherAI	6B	825GB
2021	Yuan 1.0 (S. Wu et al. 2021)	Chinese	Inspur AI	245B	5TB
2021	MT-NLG (Smith et al. 2022)	English	Microsoft+NVIDIA	530B	>825GB (est. 1.9TB)
2021	Cedille (Müller et al. 2022)	French	Cedille AI	6B	1.1TB
2022	OPT (Large) (S. Zhang et al. 2022)	English	Meta AI	175B	800GB
2022	Diffusion-LM (X. L. Li et al. 2022)	English	Stanford	300M	**40MB**
2022	YaLM (link)	Russian+English	Yandex	100B	1.7TB
2022	UL2 (Tay et al. 2022)	English	Google	20B	Unclear

Table 1: A Selection of Large Language Models

learning rate tuning and novel parameter initialisation methods were also particularly reported aspects of GPT-2's architecture and pre-training.

2.3. Transfer Learning through LLMs

Knowledge transfer from pre-trained language models to downstream NLP tasks (such as text classification, natural language inference, text generation etc.) has been carried out in several ways. Below are descriptions of a relevant subset of such modalities[9].

Fine-Tuning Since the release of GPT-1 (Radford et al. 2018) and (at least) up until the release of GPT-3 (Brown et al. 2020), fine-tuning was the most dominant transfer learning paradigm in LLM-based NLP. It, nevertheless, continues to be a subject of active research (Sanh et al. 2022, Wei et al. 2022, Min et al. 2022). A canonical way of fine-tuning LLMs is to replace its final layer with a downstream task-specific layer (for example, a K neuron layer with soft-max activation, for a downstream K-class classification task) and update (fine-tune) parameters of the entire model on the downstream task using downstream data (typically for only a few epochs).

Few-Shot Learning In few-shot settings, the parameters of the pre-trained model are not updated as in fine-tuning. Instead, the downstream task is itself presented as an input sequence. This sequence must, in natural language, contain some $K > 0$ demonstrations of the task, each consisting of a downstream input sequence ('context') and corresponding label ('completion'), if any, followed by an instance of the task context alone, without a label. The model's output corresponding to this input sequence is taken to be the prediction or more generally, the 'task completion'. This process is repeated for all downstream instances where task completion is required.

Zero-Shot Learning Same as few-shot learning, except that $K = 0$, i.e, no demonstrations are input to the model.

2.4. The Evaluation Metric, R_3

R_K (Gorodkin 2004) is a K-class generalisation of the Mathews Correlation Coefficient. Suppose that we are given a $K \times K$ confusion matrix $\mathbf{C}$ and that we denote by $\mathbf{t} = [t_1, t_2, ..., t_K]^T$ and $\mathbf{p} = [p_1, p_2, ..., p_K]^T$ the vectors of the true and predicted number of occurrences of each class in $\mathbf{C}$. Then, R_K is computed as follows.

$$N = \mathbf{1}^T \mathbf{C} \mathbf{1}$$
$$R_K = \frac{N \operatorname{tr}(\mathbf{C}) - \mathbf{t}^T \mathbf{p}}{\sqrt{N^2 - \mathbf{p}^T \mathbf{p}} \sqrt{N^2 - \mathbf{t}^T \mathbf{t}}} \tag{4}$$

It can be shown (Gorodkin 2004) that $R_K \in [-1, 1]$. R_K is greater the better, and in particular, $R_K = 1$ represents the case where all predictions are correct. Furthermore, R_3, in particular, is the case corresponding to $K = 3$, i.e., a 3-class classification problem. It is the GLUE benchmark (A. Wang et al. 2018) metric for evaluating model performance in the NLI task on the handcrafted GLUE NLI Diagnostic dataset, which, the presenters note is class-imbalanced.

3. Data

In this project, GPT-2 was fine-tuned for the NLI task using the SNLI dataset[10] (Bowman et al. 2015). The same was downloaded from the HuggingFace `Datasets` package[11]. To test out-of-distribution (OOD) generalisation on the same task (NLI), four other benchmark datasets with a similar structure, viz. `premise-hypothesis-label` were used.

To be more precise, in all the datasets used except one, called HANS, labels took one of three values – `entailment`, `contradiction` and `neutral`. In HANS, however,

[9] Zero- and few-shot learning settings have also been formulated in other ways; the description provided in 2.3 correspond closely to 'in-context learning' settings presented by Brown et al. 2020

[10] Task and dataset covered in lab 4 of the course, hence not describing the same in much detail

[11] `https://github.com/huggingface/datasets`

Algorithm 1: AFLite

Input: Dataset $\mathscr{D} = (X, Y)$,
pre-computed representation Φ,
model family $\mathscr{M}$,
target dataset size $n \leq |\mathscr{D}|$,
number of random partitions m,
training set size $t < n$,
slice size $k \leq n$,
early-stopping threshold $\tau \in [0, 1]$
Output: Reduced dataset S
$S \leftarrow D$
while $|S| > n$ **do**
  forall $i \in S$ **do**
    Initialise multiset out-of-sample predictions $E(i) = \phi$
  for *iteration* j : $1..m$ **do**
    Randomly partition S into $(T_j, S \setminus T_j)$ such that $|S \setminus T_j| = t$
    Train classifier $\mathscr{L} \in \mathscr{M}$ on $\{(\Phi(x), y) \mid (x, y) \in S \setminus T\}$
    `//`$\mathscr{L}$ `is typically a linear classifier//`
    forall $i = (x, y) \in T_j$ **do**
      $E(i) \leftarrow E(i) \uplus \mathscr{L}(\Phi(x))$ `//`$\mathscr{L}(\Phi(x))$ `is` $\mathscr{L}$`'s prediction given input` x `and representation` Φ`//`
  forall $i = (x, y) \in S$ **do**
    Compute $\tilde{p}(i) \leftarrow |\{\hat{y} \in E(i) \mid \hat{y} = y\}| / |E(i)|$
  Assign as S' up to k instances $i = (x, y) \in S$ with the highest $\tilde{p}(i)$, subject to $\tilde{p}(i) > \tau$
  $S \leftarrow S \setminus S'$
  if $|S'| < k$ **then**
    break
return S

the labels only take one of two values – `entailment` and `non-entailment`, the latter corresponding to instances that would have either been labelled `contradiction` or `neutral` in the other datasets. Under the assumption that NLI training prepares models to recognise `entailment`, it is natural to expect that learning from NLI training transfers to the task posed by datasets similar to the HANS dataset, called Recognising Textual Entailment (RTE) (Dagan et al. 2006).

By way of pre-processing, all premise-hypothesis pairs were concatenated with pipe ('|') separation and tokenised for GPT-2 compatibility. The maximum length of tokenised sequences was set to 128 tokens, to facilitate faster training; this resulted in the removal of at most 8% of instances across datasets[12]. Shorter sequences were padded on the left using GPT2's end-of-sequence token.

HANS (McCoy et al. 2019) was designed to identify the learning of three specific structural heuristics – *Lexical Overlap* (Lex.), *Subsequence* (Subseq.), and *Constituent* (Constit.) – that can adversely impact model generalisation. The authors define lexical overlap, for instance, as a heuristic which encourages the model to learn that a premise entails any hypothesis constructed entirely from words in the premise. This heuristic wouldn't work, for instance, in the example they provide – Premise. "The doctor was paid by the actor"; Hypothesis. "The doctor paid the actor". The HANS dataset was also downloaded from the HuggingFace `Datasets` package.

NLI Diagnostics (A. Wang et al. 2018) were handcrafted to test model performance on several fine-grained semantic categories, grouped under the headings *Lexical Semantics* (LxS), *Predicate-Argument Structure (PAS)*, *Logic*, *Knowledge* (Knowl.) and *Domain*. In the project, evaluations were made on all groupings except *Domain*, as Le Bras et al. 2020 do, for comparing results. This dataset was downloaded from the GLUE Diagnostics dataset page[13].

Stress Tests for NLI (Naik et al. 2018) are a suite of three kinds of tests. *Competence* (Comp.) tests evaluate model performance in the presence of quantitative and antonymy relations, *Distraction* (Distr.) tests, in the presence of shallow distractions such as presence of negation words and *Noise*, in the presence of noisy data such as typos. Corresponding datasets were downloaded from the GitHub link cited in the paper[14].

Adversarial NLI (ANLI) (Nie et al. 2020) form another suite of three NLI datasets (rounds) designed to each be progressively more difficult than the previous. The premises in each dataset were scraped from Wikipedia and other corpora, whilst the hypotheses and labels were human generated, through a novel human- and model-in-the-loop procedure. Corresponding datasets were downloaded from the HuggingFace `Datasets` package.

4. Method

We may recall that the project objective is to check whether there are representation biases in the SNLI dataset with respect to GPT-2 (specifically, GPT2-small (117M parameters) and GPT2-medium (345M parameters), due to resource constraints), by means of AFLite. I take the following approach: for each of these two LLMs, I generate 5 filtered datasets (candidates) using AFLite, each generated using a different random seed. I then evaluate the performance of the LLMs fine-tuned on the NLI task using candidate datasets and compare the same with that of two baseline models for each LLM. For additional comparison, I also include results from Le Bras et al. 2020 concerning RoBERTa-large performance on the NLI task using AFLite filtered datasets and corresponding baselines. Below are details concerning each of the steps mentioned above.

Candidate Dataset Generation A random 10% subset, viz. *warmup*, of the SNLI training dataset was extracted and used for fine-tuning a given LLM on the NLI task. The network corresponding to the resulting model, with its output layer removed, was taken to be the feature representation Φ corresponding to the given LLM. AFLite was then run with the following inputs. $\mathscr{D}$ was taken to be the unfiltered SNLI training dataset, $\mathscr{M}$, linear classifiers with soft-max activation, n, 195k, m, 30, t, 50k, k, 100k and τ, 0.75. The values of n, t and τ were chosen to match the out-of-distribution experiments of Le Bras et al. 2020, so as to have some oppor-

[12] See A for details on number of exclusions made

[13] `https://gluebenchmark.com/diagnostics`

[14] `https://abhilasharavichander.github.io/NLI_StressTest`

		HANS (RTE Task)			**NLI-Diagnostics** (NLI Task)				**Stress Tests** (NLI Task)		
		Lex.	*Subseq.*	*Constit.*	*Knowl.*	*Logic*	*PAS*	*LxS.*	*Comp.*	*Distr.*	*Noise*
GPT2-small	$\mathscr{D}$	**50.0**	50.0	**50.0**	37.0	**42.0**	52.8	44.3	**31.1**	48.7	**56.0**
	$\mathscr{D}_{190k}$	**50.0**	50.0	49.7	38.0	41.7	55.7	44.0	16.7	45.2	49.6
	$\mathscr{D}(\phi_{GPT2-small})$	$50.0_{0.0}$	$\mathbf{50.0}_{0.0}$	$50.0_{0.1}$	$\mathbf{39.4}_{2.4}$	$41.4_{1.7}$	$\mathbf{56.0}_{3.2}$	$\mathbf{44.4}_{1.4}$	$25.2_{4.0}$	$\mathbf{49.4}_{1.1}$	$54.5_{0.8}$
GPT2-medium	$\mathscr{D}$	**53.3**	**50.6**	50.2	**43.0**	**46.1**	**61.8**	**53.5**	**42.1**	**58.1**	**64.7**
	$\mathscr{D}_{190k}$	50.0	49.9	**51.3**	38.7	43.4	59.0	49.2	28.8	51.1	57.4
	$\mathscr{D}(\phi_{GPT2-medium})$	$50.3_{0.7}$	$50.0_{0.3}$	$50.4_{0.4}$	$40.5_{2.0}$	$42.4_{0.8}$	$56.8_{1.5}$	$45.8_{2.0}$	$35.3_{5.2}$	$53.9_{1.7}$	$59.1_{1.8}$
RoBERTa-large	$\mathscr{D}$	$88.4_{2.2}$	$28.2_{3.4}$	$21.7_{7.1}$	$51.8_{1.6}$	$57.8_{1.7}$	$\mathbf{72.6}_{1.3}$	$65.7_{1.9}$	$77.9_{2.5}$	$\mathbf{73.5}_{2.9}$	$\mathbf{79.8}_{0.8}$
(results from	$\mathscr{D}'_{190k}$	$56.6_{14.7}$	$19.6_{5.6}$	$13.8_{2.9}$	$\mathbf{56.4}_{0.8}$	$53.9_{1.5}$	$71.2_{1.1}$	$65.6_{1.7}$	$68.4_{3.0}$	$73.0_{3.0}$	$78.6_{0.4}$
Le Bras et al. 2020)	$\mathscr{D}(\phi_{RoBERTa-large})$	$\mathbf{94.1}_{3.5}$	$\mathbf{46.3}_{6.0}$	$\mathbf{38.5}_{15.2}$	$53.9_{1.6}$	$\mathbf{58.7}_{1.2}$	$69.9_{0.9}$	$\mathbf{66.5}_{1.7}$	$\mathbf{79.1}_{1.0}$	$72.0_{1.8}$	$79.5_{0.4}$

Table 2: Zero-Shot Accuracy of Candidate and Baseline models in OOD datasets

tunity for comparison of results[15], [16]. The value of m was reduced from 64, as in Le Bras et al. 2020, to 30, and value of k was similarly increased from 10k to 100k for faster implementation in light of resource constraints. A total of 10 candidate datasets were generated, 5 for each LLM, each with a different random seed.

Baselines For each given LLM, two of the following baseline models were developed. The first was the LLM fine-tuned on the (complete) SNLI training dataset, and the second, that fine-tuned on a random subset of the SNLI training dataset whose size (number of instances) was approximately the same as that of the AFLite filtered candidates.

Evaluation A total of fourteen models were evaluated - 10 resulting from fine-tuning the LLMs on the candidate datasets (candidate models) and 4 baseline models. Evaluation consisted of experiments wherein each model's performance, as indicated by classification accuracy and R_K, was measured on the (complete) in-distribution SNLI test dataset and the out-of-distribution datasets listed in §3.

All evaluations except those performed on the ANLI datasets were zero-shot evaluations. For ANLI datasets, each of the 14 models were further fine-tuned on the ANLI training datasets before evaluation, for each of the three ANLI rounds. Taken together, fourteen test tasks were developed in all; a single in-distribution test task, 3 zero-shot test tasks, using the HANS dataset (one for each heuristic), 4, using the NLI diagnostics dataset, 3 more, using the Stress Test datasets and 3 test tasks involving further fine-tuning, using the ANLI datasets.

Implementation Details The following is a list of key Python packages/classes used in this project: the `GPT2TokenizerFast` and `DataCollatorWithPadding` classes from the HuggingFace package `transformers` were used for data pre-processing; the `GPT2ForSequenceClassification`[17] class from the HuggingFace package `transformers` was used for model training using PyTorch. As in Le Bras et al. 2020, all training procedures involved three epochs of training, Adam optimiser[18] with default hyper-parameters, a learning rate of 10^{-5}, categorical cross-entropy loss[19] as the training loss function and a single RTX 3090 GPU with 24 GB RAM.

Depending upon procedure involved, training data were batched as groups of 32, 64, 92, or 128 sequences; details in Appendix B. The choice of batch sizes used were initially made for complete alignment with Le Bras et al. 2020, who use a constant batch size of 92 for all training procedures; in this project, batch sizes of 32 and 64 had to be chosen for certain procedures because of GPU RAM constraints; finally, batch size of 128 was chosen only for one procedure, and this was done to speed up computations.

5. Results

The performance of GPT2-small, GPT2-medium, and where possible, RoBERTa-large fine-tuned for the NLI (RTE) test tasks using various datasets are reported in tables 2, 3, 4 and 5. Performance metrics include classification accuracy and R_K (R_2 for RTE tasks and R_3 for NLI tasks). $\mathscr{D}$, $\mathscr{D}_{190k}$ and $\mathscr{D}_{\Phi_{LLM}}$ are used as abbreviations for the full (unfiltered) SNLI dataset, a 190k size random subset of the same, and the AFLite filtered SNLI datasets, respectively, thereby indicating which dataset was primarily used to fine-tune the corresponding LLM[20].

Model performance was tested on in-distribution and out-of-distribution datasets as described in §3. The means and standard deviations of the performance of candidate models (i.e., LLMs fine-tuned on candidate datasets, viz. the AFLite filtered SNLI datasets), as measured in each test task are respectively reported in standard and subscript font size. For each test task and LLM combination, the best observed performance is highlighted in bold face. Finally, all RoBERTa-large related performance measures reported are as done in Le Bras et al. 2020.

In none of the 13 out-of-distribution tests were significant improvements observed in performances of GPT2-based

[15] Le Bras et al. 2020 use $n = 182$k for out-of-distribution experiments and $n = 92$k for in-distribution experiments; in this project, the $t = 50$k and $\tau = 0.75$ remained the same in both kinds of experiments

[16] See §6 for a discussion on the issues pertinent to comparing results in this project versus those in Le Bras et al. 2020

[17] `transformers.GPT2ForSequenceClassification`

[18] `torch.optim.Adam.html`

[19] `torch.nn.CrossEntropyLoss`

[20] As indicated in §4, tests involving the ANLI datasets involved further fine-tuning using the ANLI training datasets

		HANS (RTE Task)			**NLI-Diagnostics** (NLI Task)				**Stress Tests** (NLI Task)		
		Lex.	*Subseq.*	*Constit.*	*Knowl.*	*Logic*	*PAS*	*LxS.*	*Comp.*	*Distr.*	*Noise*
GPT2-small	$\mathscr{D}$	**0.01**	0.00	**−0.01**	0.06	**0.12**	0.19	**0.14**	−0.15	0.25	**0.35**
	$\mathscr{D}_{190k}$	0.00	−0.01	−0.02	**0.10**	0.11	0.22	0.11	**−0.03**	0.20	0.26
	$\mathscr{D}(\phi_{GPT2-small})$	$0.00_{0.00}$	$\mathbf{0.00}_{0.02}$	$-0.01_{0.01}$	$0.10_{0.02}$	$0.12_{0.03}$	$\mathbf{0.24}_{0.05}$	$0.13_{0.02}$	$-0.15_{0.03}$	$\mathbf{0.25}_{0.01}$	$0.32_{0.01}$
GPT2-medium	$\mathscr{D}$	**0.18**	**0.07**	0.03	**0.14**	**0.19**	**0.34**	**0.29**	−0.11	**0.38**	**0.47**
	$\mathscr{D}_{190k}$	0.01	−0.03	**0.11**	0.11	0.14	0.28	0.21	−0.10	0.30	0.37
	$\mathscr{D}(\phi_{GPT2-medium})$	$0.03_{0.04}$	$-0.01_{0.03}$	$0.03_{0.02}$	$0.11_{0.05}$	$0.13_{0.01}$	$0.25_{0.02}$	$0.16_{0.03}$	$\mathbf{-0.10}_{0.07}$	$0.32_{0.02}$	$0.40_{0.02}$

Table 3: Zero-Shot R_K of Candidate and Baseline models in Out-Of-Distribution datasets

candidate models vis-à-vis corresponding baselines, nor were significant reductions in performance observed in the in-distribution test.

6. Discussion

6.1. Results

Whereas Le Bras et al. 2020 report modest to significant improvements in classification accuracy in most out-of-distribution tests and a 30% points reduction in classification accuracy in the in-distribution test for RoBERTa-large, after training data filtering using AFLite, no such effects were observed for GPT2-small and -medium. Below, I analyse this finding from three different perspectives.

LLM Size A comparison of results observed for GPT2-small (117M parameters) and -medium (345M parameters) LLMs allows for some insight into how LLM size may impact the performance of candidate and baseline models developed in this project. As in most deep-learning applications, however, the statistical significance of these comparisons may be called into question. §§6.2 offers a brief discussion on this topic. Nevertheless, we can notice that when fine-tuned on the full (unfiltered) SNLI training dataset, $\mathscr{D}$ (baseline 1), GPT2-medium delivered better performance in all the 10 zero-shot out-of-distribution tests and the in-distribution test. Moreover, in two of the three test tasks involving further fine-tuning using ANLI training datasets, viz. rounds 1 and 3, GPT2-medium outperformed GPT2-small. Finally, GPT2-medium outperformed GPT2-small in the in-distribution test task as well.

		Accuracy	**R3**
GPT2-small	$\mathscr{D}$	**85.6**	**0.78**
	$\mathscr{D}_{190k}$	79.5	0.69
	$\mathscr{D}(\phi_{GPT2-small})$	$83.9_{0.1}$	$0.76_{0.00}$
GPT2-medium	$\mathscr{D}$	**89.4**	**0.84**
	$\mathscr{D}_{190k}$	86.4	0.80
	$\mathscr{D}(\phi_{GPT2-medium})$	$86.8_{0.6}$	$0.80_{0.01}$
RoBERTa-large (results from Le Bras et al. 2020)	$\mathscr{D}$	**92.6**	-
	$\mathscr{D}_{92k}$	88.3	-
	$\mathscr{D}(\phi_{92k\ RoBERTa-large})$	62.6	-

Table 4: Zero-Shot In-Distribution Accuracy and R_3 of Fine-Tuned Candidate and Baseline models

Similarly, when fine-tuned on (the same) random subset of the SNLI training dataset, $\mathscr{D}_{190k}$ (baseline 2), GPT2-medium performed better in 8 out of 10 zero-shot out-of-distribution test tasks, one of 3 ANLI test tasks, and the in-distribution test task. Moreover, results pertaining to the candidate models, i.e., LLMs fine-tuned on AFLite filtered SNLI training datasets, $\mathscr{D}_{\phi_{LLM}}$, show that GPT2-medium performance, on average[21] was better than GPT2-small's in 9 out of 10 zero-shot out-of-distribution test tasks, one of 3 ANLI test tasks, and the in-distribution test task.

These results do *not* provide much evidence against the view that language model size improves performance in natural language understanding tasks, one that has been widely held since Radford et al. 2019. Perhaps more pertinently, if AFLite (as run with the hyperparameter choices made in this project) has in fact identified and removed representation biases in the SNLI training dataset with respect to GPT2-small and -medium, one cannot say that the larger GPT2-medium LLM was anymore overfit on SNLI training data than GPT2-small, despite being $\approx 3\times$ larger in size.

Performance of GPT2-medium vs. RoBERTa-large Ascertaining the specific contributions of an LLM's pre-training task (e.g., masked language modelling versus unmasked language modelling tasks), various architectural, hyperparameter, and pre-training data choices would involve expensive ablation studies. Indeed GPT2 and RoBERTa models differ from each other on a number of these aspects, and an ideal comparison should isolate the impact of each of these aspects on performance. This project, nevertheless, does offer some opportunity to compare the performance of GPT2 and RoBERTa (Y. Liu et al. 2019), at large, in NLI/RTE tasks, since GPT2-medium and RoBERTa-large have 345M and 355M learned parameters respectively, allowing for some control over one such aspect, viz. LLM size.

As mentioned earlier, RoBERTa-large results cited in this paper are those produced by Le Bras et al. 2020, who measure performance only via classification accuracy. Note, however, that constrained experimental design in this project and brevity in the description of the design of experiments in Le Bras et al. 2020 have given way to some measurement issues discussed in §§6.2. Reverting to the said comparison, nevertheless, we can see from the results that RoBERTa-large, when fine-tuned with the full (unfiltered) SNLI training dataset, $\mathscr{D}$ (baseline 1), outperforms the similarly fine-tuned GPT2-medium model significantly in all test tasks except two HANS RTE tasks.

[21] Average performance computed via 5 different random seeds, which each generated a different AFLite filtered SNLI training dataset

		ANLI Accuracy			ANLI R_3		
		Round 1	*Round 2*	*Round 3*	*Round 1*	*Round 2*	*Round 3*
GPT2-small	$\mathscr{D}$	**32.1**	32.3	**34.9**	−0.03	−0.02	**0.02**
	$\mathscr{D}_{190k}$	31.3	**33.3**	34.5	−0.04	**0.00**	0.01
	$\mathscr{D}(\phi_{GPT2-small})$	$31.8_{1.0}$	$31.4_{0.6}$	$34.3_{0.5}$	$\mathbf{-0.03}_{0.01}$	$-0.03_{0.01}$	$0.01_{0.01}$
GPT2-medium	$\mathscr{D}$	**32.4**	30.1	**36.8**	**−0.01**	−0.05	**0.05**
	$\mathscr{D}_{190k}$	29.1	**31.7**	34.8	−0.06	**−0.03**	0.02
	$\mathscr{D}(\phi_{GPT2-medium})$	$30.4_{1.3}$	$31.4_{1.2}$	$35.8_{1.2}$	$-0.05_{0.02}$	$-0.03_{0.02}$	$0.04_{0.02}$
RoBERTa-large (results from Le Bras et al. 2020)	$\mathscr{D}$	58.5	48.3	50.1	-	-	-
	$\mathscr{D}'_{190k}$	-	-	-	-	-	-
	$\mathscr{D}(\phi_{RoBERTa-large})$	**65.1**	**49.1**	**52.8**	-	-	-

Table 5: ANLI Accuracy and R_3 of Fine-Tuned Candidate and Baseline models

Barring the issue of disparity in the random initialisation of the classification layer (head) of the two models before fine-tuning on the full SNLI training dataset, an issue which may be accounted for to a reasonable extent by reports of the standard deviations in RoBERTa-large performance, there is no uncertainty in the performance evaluations in this context. One may thus infer that RoBERTa-large generalises better both on the in- and out-of-distribution NLI tasks than GPT2-medium *without any intervention*.

RoBERTa-large similarly performs better than GPT2-medium when fine-tuned on a random 190k sized subset of the SNLI training dataset, $\mathscr{D}_{190k}$ (baseline 2) in the zero-shot out-of-distribution test tasks designed. Le Bras et al. 2020 do not present RoBERTa-large performance in ANLI test tasks when fine-tuned on the random subset and for the in-distribution test task, RoBERTa-large was fine-tuned on a 92k sized subset, so as to compare it with an AFLite filtered subset of size 92k; in this project, 92k sized subsets were not computed due to resource constraint. Finally, and possibly most pertinently, AFLite seems to further enhance RoBERTa-large performance on several out-of-distribution NLI test tasks, which it does not do for GPT2-medium. As mentioned earlier, these comparisons do not account for the measurement issues set out in §§6.2.

Performance as measured by Classification Accuracy and R_K The relative performance GPT2-based NLI models remained consistent irrespective of the choice of performance measure - classification accuracy or R_K - on all the test tasks designed in this project. This finding indicates that model performance was *not* marred by the class imbalance inherent in the various datasets used in the project.

6.2. Limitations

Hyperparameter Choices Perhaps the most crippling limitation in this project was that I could not conduct an extensive search for AFLite hyper-parameters because of how much compute a single run of AFLite costs given a single set of hyper-parameters. On the hardware (including a relatively powerful GPU) used in this project (specified in §4), a single AFLite run (also as specified in §4) required ≈ 70 hours (3 days) for GPT2-medium. Given AFLite's success on RoBERTa-large (Le Bras et al. 2020), one cannot rule out this limitation as the cause for AFLite's failure to identify representation biases in GPT2-small and -medium.

Model Performance Measurement Proper estimation of the performances of models whose fine-tuning (or more generally, training) exposes them to one or more random effects requires us to control for these effects. Not doing so opens up performance estimates to reasonable doubt that the same were obtained just by chance. When models are small and it is inexpensive to test model performance in multiple tasks, experimental design can be so nuanced as to control for these effects without significant cost. Alternatively, if there is expert knowledge of both the tasks (likelihood functions) and the model (parameter priors), one may be able to use Bayesian principles and design experiments accounting for the random effects even if a single test were expensive without significant cost. In the NLI tasks described in this project, because of the large size of the test datasets and models and the high compute costs involved in testing, a number of random effects remain uncontrolled for in this project (as also in Le Bras et al. 2020, and more broadly, in several Deep Learning applications).

To offer some perspective on how much of an issue this really is, in this project, I list below several random effects the LLMs in this project were subject to. Firstly, all models in this report involved random initialisation of the classification head of the LLMs before fine-tuning. Whereas in my implementation, I do *seed* these random initialisation procedures consistently for all the GPT-2 based models, Le Bras et al. 2020 do not share what random seeds they used for the RoBERTa-large models (they have not published their code), opening up avenues for disparity. Secondly, as for models pre-trained using random subsets of the SNLI training dataset (baseline 2), the selection of instances to be included in the said subsets is random. Here again, similar disparities possibly arise. Finally, AFLite runs involve exposing models to more random effects, such as the random choice of the *warmup* dataset and bootstrap sampling of the training dataset for filtering.

Consistency in random seed settings, which I do maintain in this project, helps compare performance of models exposed to the random effects seeded in the same manner, but do not present an unbiased (statistically valid) estimation of model performance. Failing to design experiments that con-

trol for these random effects, notwithstanding the issue of cost (compute and time), therefore calls into question the statistical significance of the results I report in this project. Furthermore, note that by way of accounting for the various random effects listed above, Le Bras et al. 2020 run their experiments using 5 random seeds. Unfortunately, because there are multiple random effects at play simultaneously, using 5 random seeds and reporting standard deviations in performance measures does not control for all of them at once, either.

6.3. Related Work

AFLite for Adversarially Robust NLI This project uses AFLite exactly as presented by Sakaguchi et al. 2021. Le Bras et al. 2020 investigate how to use AFLite for adversarially robust NLI. Their work, however, is limited to RoBERTa-large insofar as out-of-distribution tests are concerned and to BERT-large (Devlin et al. 2019), RoBERTa-large and the ESIM+ELMO (Peters et al. 2018), insofar as the in-distribution test is concerned. This project expands upon their work and focusses primarily on GPT2-small and -medium in the context of adversarially robust NLI through AFLite. Furthermore, this project involved a completely original implementation of AFLite (neither Sakaguchi et al. 2021 nor Le Bras et al. 2020 publish their implementations of AFLite), which has been made publicly available.

Alternatives for Adversarially Robust NLI using LLMs Bahng et al. 2020 propose 'ReBias', a framework designed to train de-biased representations by rewarding those that are different from representations that are designed to be biased. Y. Li et al. 2019 propose 'REPAIR' a framework to remove representation bias in datasets by weighting instances in training datasets such that instances which are easy to classify using a given representation are penalised. Clark et al. 2019 propose a method for developing robust models by learning how much to trust multiple naive models designed to learn only from biased datasets (such as only the *premise* or *hypothesis* in the SNLI dataset).

7. Conclusion

A great deal of resources continue to pour into the development of large language models (LLMs) for solving Natural Language Understanding (NLU) problems. However, LLMs struggle deal with out-of-distribution NLU problems where human performance remains robust. Spurious artefacts (biases) inherent in training datasets are ostensibly picked up by models, particularly those such as LLMs, which are heavily parameterised, making them specialists at solving tasks that are extremely similar to that they were trained on and less adept at solving tasks, which, for the models may be adversarial, but, which, for humans are no different than the training task in essence.

Imbuing models with human-level intelligence thus requires the development of adversarially robust models. In this project, I attempted to make a particular LLM, GPT-2, robust to adversarial natural language inference (NLI) datasets. Given resource constraints (compute and time), however, I was unable to achieve that goal entirely. I was, however, able to make publicly available a software implementation of my experiments and outline a way forward that could involve building upon this project and/or other promising research directions.

References

Bahng, Hyojin et al. (2020). "Learning De-biased Representations with Biased Representations". In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, pp. 528–539. URL: https://proceedings.mlr.press/v119/bahng20a.html.

Bender, Emily M. et al. (2021). "On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?" In: *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. FAccT '21. Virtual Event, Canada: Association for Computing Machinery, pp. 610–623. ISBN: 9781450383097. DOI: 10.1145/3442188.3445922. URL: https://doi.org/10.1145/3442188.3445922.

Bowman, Samuel R. et al. (Sept. 2015). "A large annotated corpus for learning natural language inference". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, pp. 632–642. DOI: 10.18653/v1/D15-1075. URL: https://aclanthology.org/D15-1075.

Brown, Tom et al. (2020). "Language Models are Few-Shot Learners". In: *Advances in Neural Information Processing Systems* 33, pp. 1877–1901.

Clark, Christopher, Mark Yatskar, and Luke Zettlemoyer (2019). "Don't Take the Easy Way Out: Ensemble Based Methods for Avoiding Known Dataset Biases". 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)*. Hong Kong, China: Association for Computational Linguistics, pp. 4069–4082. DOI: 10.18653/v1/D19-1418. URL: https://aclanthology.org/D19-1418.

Dagan, Ido, Oren Glickman, and Bernardo Magnini (2006). "The PASCAL Recognising Textual Entailment Challenge". en. In: *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*. Ed. by Joaquin Quiñonero-Candela et al. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, pp. 177–190. ISBN: 978-3-540-33428-6. DOI: 10.1007/11736790_9.

Devlin, Jacob et al. (June 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)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://aclanthology.org/N19-1423.

Goodfellow, Ian, Jonathon Shlens, and Christian Szegedy (2015). "Explaining and Harnessing Adversarial Exam-

ples". In: *International Conference on Learning Representations*. URL: http://arxiv.org/abs/1412.6572.

Gorodkin, J. (2004). "Comparing two K-category assignments by a K-category correlation coefficient". In: *Computational Biology and Chemistry* 28.5, pp. 367–374. ISSN: 1476-9271. DOI: https://doi.org/10.1016/j.compbiolchem.2004.09.006. URL: https://www.sciencedirect.com/science/article/pii/S1476927104000799.

Jin, Di et al. (2020). "Is BERT Really Robust? A Strong Baseline for Natural Language Attack on Text Classification and Entailment". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 34. 05, pp. 8018–8025.

Le Bras, Ronan et al. (2020). "Adversarial Filters of Dataset Biases". In: *International Conference on Machine Learning*. PMLR, pp. 1078–1088.

Li, Xiang Lisa et al. (2022). "Diffusion-LM Improves Controllable Text Generation". In: *arXiv preprint arXiv:2205.14217*.

Li, Yi and Nuno Vasconcelos (2019). "REPAIR: Removing Representation Bias by Dataset Resampling". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Liu, Yinhan et al. (2019). "Roberta: A Robustly Optimized BERT Pretraining Approach". In: *arXiv preprint arXiv:1907.11692*.

McCoy, Tom, Ellie Pavlick, and Tal Linzen (July 2019). "Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 3428–3448. DOI: 10.18653/v1/P19-1334. URL: https://aclanthology.org/P19-1334.

Min, Sewon et al. (July 2022). "MetaICL: Learning to Learn In Context". In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Seattle, United States: Association for Computational Linguistics, pp. 2791–2809. URL: https://aclanthology.org/2022.naacl-main.201.

Müller, Martin and Florian Laurent (2022). "Cedille: A large autoregressive French language model". In: *arXiv preprint arXiv:2202.03371*.

Naik, Aakanksha et al. (Aug. 2018). "Stress Test Evaluation for Natural Language Inference". In: *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, pp. 2340–2353. URL: https://aclanthology.org/C18-1198.

Nie, Yixin et al. (July 2020). "Adversarial NLI: A New Benchmark for Natural Language Understanding". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 4885–4901. DOI: 10.18653/v1/2020.acl-main.441. URL: https://aclanthology.org/2020.acl-main.441.

Niven, Timothy and Hung-Yu Kao (July 2019). "Probing Neural Network Comprehension of Natural Language Arguments". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 4658–4664. DOI: 10.18653/v1/P19-1459. URL: https://aclanthology.org/P19-1459.

Peters, Matthew E. et al. (June 2018). "Deep Contextualized Word Representations". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 2227–2237. DOI: 10.18653/v1/N18-1202. URL: https://aclanthology.org/N18-1202.

Pruthi, Danish, Bhuwan Dhingra, and Zachary C. Lipton (July 2019). "Combating Adversarial Misspellings with Robust Word Recognition". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 5582–5591. DOI: 10.18653/v1/P19-1561. URL: https://aclanthology.org/P19-1561.

Radford, Alec et al. (2018). "Improving Language Understanding by Generative Pre-Training". In: *Technical Report, OpenAI*.

Radford, Alec et al. (2019). "Language Models are Unsupervised Multitask Learners". In: *OpenAI blog* 1.8, p. 9.

Sakaguchi, Keisuke et al. (Aug. 2021). "WinoGrande: An Adversarial Winograd Schema Challenge at Scale". In: *Commun. ACM* 64.9, pp. 99–106. ISSN: 0001-0782. DOI: 10.1145/3474381. URL: https://doi.org/10.1145/3474381.

Sanh, Victor et al. (2022). "Multitask Prompted Training Enables Zero-Shot Task Generalization". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=9Vrb9D0WI4.

Shoeybi, Mohammad et al. (2019). "Megatron-LM: Training Multi-Billion Parameter Language Models Using Model Parallelism". In: *arXiv preprint arXiv:1909.08053*.

Smith, Shaden et al. (2022). "Using DeepSpeed and Megatron to Train Megatron-Turing NLG 530B". In: *arXiv preprint arXiv:2201.11990*.

Szegedy, Christian et al. (2014). "Intriguing properties of neural networks". In: *International Conference on Learning Representations*. URL: http://arxiv.org/abs/1312.6199.

Tay, Yi et al. (2022). "Unifying Language Learning Paradigms". In: *arXiv preprint arXiv:2205.05131*.

Vaswani, Ashish et al. (2017). "Attention Is All You Need". In: *Advances in neural information processing systems* 30.

Wang, Alex et al. (Nov. 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*. Brussels, Belgium: Association for Computational Linguistics, pp. 353–355. DOI: 10.18653/v1/W18-5446. URL: https://aclanthology.org/W18-5446.

Wang, Alex et al. (2019). "SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Sys-

tems". In: *Advances in Neural Information Processing Systems* 32.

Wang, Ben (May 2021). *Mesh-Transformer-JAX: Model-Parallel Implementation of Transformer Language Model with JAX*. https://github.com/kingoflolz/mesh-transformer-jax.

Wei, Jason et al. (2022). "Finetuned Language Models are Zero-Shot Learners". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=gEZrGCozdqR.

Wu, Shaohua et al. (2021). "Yuan 1.0: Large-scale pre-trained language model in zero-shot and few-shot learning". In: *arXiv preprint arXiv:2110.04725*.

Xiao, Dongling et al. (July 2020). "ERNIE-GEN: An Enhanced Multi-Flow Pre-training and Fine-tuning Framework for Natural Language Generation". In: *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*. Ed. by Christian Bessiere. Main track. International Joint Conferences on Artificial Intelligence Organization, pp. 3997–4003. DOI: 10.24963/ijcai.2020/553. URL: https://doi.org/10.24963/ijcai.2020/553.

Zang, Yuan et al. (July 2020). "Word-level Textual Adversarial Attacking as Combinatorial Optimization". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 6066–6080. DOI: 10.18653/v1/2020.acl-main.540. URL: https://aclanthology.org/2020.acl-main.540.

Zhang, Susan et al. (2022). "OPT: Open Pre-trained Transformer Language Models". In: *arXiv preprint arXiv:2205.01068*.

A. Appendix 1: Exclusions due to 128 token sequence length limit

Dataset	*Fold*	*Split within Fold*	*# Instances Excluded*	*% Instances Excluded*	*# Instances Remaining*
ANLI	Round 1	Train	446	2.7%	16,072
	Round 1	Test	23	2.3%	977
	Round 2	Train	944	2.1%	44,008
	Round 2	Test	18	1.8%	982
	Round 3	Train	7 468	8.0%	85,882
	Round 3	Test	90	8.1%	1,021
Stress Test	Heuristic	Test	396	1.7%	22,898
	Distraction	Test	1,060	0.4%	263,940
	Noise	Test	1,070	0.3%	355,596

B. Appendix 2: Size of Training Batches

LLM	**Procedure**	**Batch Size**
GPT2-small	Training of Baselines	92
	AFLite: Warmup (obtaining Φ)	92
	AFLite: Filtering	128
	Finetuning with ANLI training datasets	64
GPT2-medium	All procedures except AFLite filtering	32
	AFLite: Filtering	128
RoBERTa-large (Le Bras et al. 2020)	All procedures	92