Article review & experiments

Pre-Training Small Base LMs With Fewer Tokens

Надежда Анисимова
Мария Бородкина
Георгий Гаврюшов

Репозиторий с экспериментами

Май 2024

Или рецепт, как приготовить свою маленькую LM, не сильно потеряв в качестве

Статья

Репозиторий статьи

Предисловие

Какие Этапы Превращают Простую Нейронку В LLM?

Pre-Training

Процесс обучения модели на огромном корпусе текста (миллиарды слов), чтобы модель выучила структуру языка, грамматику, общие факты о жизни.

Сравнимо с обучением ребенка другому языку, посредством чтения большого количества литературы, без глубокого понимания определенных тем

Fine-Tuning

Процесс идет после pre-training, направлен на дообучение модели на узкую специализацию для определенных задач.

Сравнимо с обучением ребенка определенной области, например биологии, уже после того, как им был изучен сам язык

Данные

Огромный корпус (например Википедия)

Меньший датасет под определенную задачу (например, юридические тексты)

Зачем Нужна Small LM?

Маленькие языковые модели обеспечивают **высокую адаптивность и быстроту реагирования**, что важно для приложений в реальном времени. Их меньший размер **снижает задержку**, делая их идеальными для обслуживания клиентов через искусственный интеллект и анализ данных в реальном времени.

Они обладают **меньшими требованиями к вычислительным ресурсам** и могут быть легче интегрированы в ограниченные среды, что делает их более экономичными и доступными для различных приложений

Что Предлагают Авторы Статьи?

Простой и дешевый способ разработки **Small base LM** для последубщего дообучения под определенную задачу

Inheritune

Этапы:

1. Наследовать несколько блоков трансформеров большой **LLM**

2. Обучить меньшую модель на подмножестве (0,1%) необработанных данных на которых обучалась большая модель

Рассмотрены Две Ситуации

Наличие **небольшой части данных** для предварительного обучения вместе с существующей большой LM

Предобучена модель:

1.5B параметров
1B токенов данных
На **1 A6000** GPU
Меньше чем за **12** часов

На основе **OpenLLamA-3B**

Получение еще меньшей LM **при наличие полного датасета** для предобучения большой LM

Эксперимены с GPT2-large GPT2-medium

GPT2-large:
50% слоев, 45% параметров

GPT2-medium:
33% слоев, 28% параметров

без потери на валидационном лоссе

General Setup

1. Inheritune V1

Algorithm 1 Inheritune

Require: Reference model $\mathcal{M}_{ref}$, a subset $\hat{\mathcal{D}}_{\text{train}}$ from $\mathcal{D}_{\text{train}}$ used to train $\mathcal{M}_{ref}$, the number of layers n we want in our final model, number of epochs E, and training hyper-parameters.

1: Let k be the number of transformer blocks in $\mathcal{M}_{ref}$, with the corresponding parameters being $\theta_0, \theta_1, \ldots, \theta_{k-1}$.

2: **Initialize:** target LM θ_{tgt} with the first n layers $\theta_0, \theta_1, \ldots, \theta_{n-1}$, as well as the token embedding and prediction head layers of $\mathcal{M}_{ref}$.

3: **Train** the target LM from this initialization for E epochs using the given hyper-parameters.

Имеем:

Mref - модель из k слоев
θref = {θ0 , θ1 , . . . , θk−1 } - параметров
Обученная на Dtrain, однако в доступе лишь его часть - D̂train
Подмножество рандомное

Шаг 1:

Инициализация Mtgt посредством **наследования первых n слоев от Mref**
Веса Mtgt: {θ0 , θ1 , . . . , θn−1 }
Блок предсказаний и эмбединги токенов также унаследованы

Шаг2:

Обучение Mtgt в течении E эпох на части данных - D̂train

Experimental Setup

1. Данные

Исходный датасет
Redpajama v1
1 триллион токенов
википедия, книги, архивы, stackexchange и др.

Использовано для эксперимента

1 миллиард токенов

Датасет собран в тех же пропорциях как предложено для LLaMa и сделано для OpenLLamA-3B

Итого датасет содержит **всего лишь 0.1% данных** для обучения OpenLLamA-3B

1 — 2 — 3 — 4

Experimental Setup

2. Модель И Обучение

Mref - модель **OpenLLamA-3B version1** из **k=26** слоев

Mtgt - полученная модель **n=k/2=13** слоев

E = **8** эпох (каждая эпоха использует все 1B токенов)

BatchSize = **131K** токенов

Models	Layers	Hidden Size	Heads
OpenLLaMA-3B	26	3200	32
OpenLLaMA-7B	32	4096	32
LLaMA2-7B	32	4096	32
GPT2-large(770M)	36	1280	20
GPT2-medium(355M)	24	1024	16

Table 1: Overview of reference models used in this study and their architectural configurations. We obtain a pre-trained OpenLLaMA-3B and we trained all GPT2 models with OpenWebText consisting of 9B tokens.

Model	**Training Data** (# tokens)
OpenLLaMA-3B v1(ref)	RedPajama(1T)
Ours-1.5B*	RedPajama (1B)
Shear-LLaMA-1.3B*	RedPajama(50B)
MPT-1.3B	RedPajama(200B)
Pythia-1.4B	The Pile(300B)
OPT-1.3B	Custom data(300B)

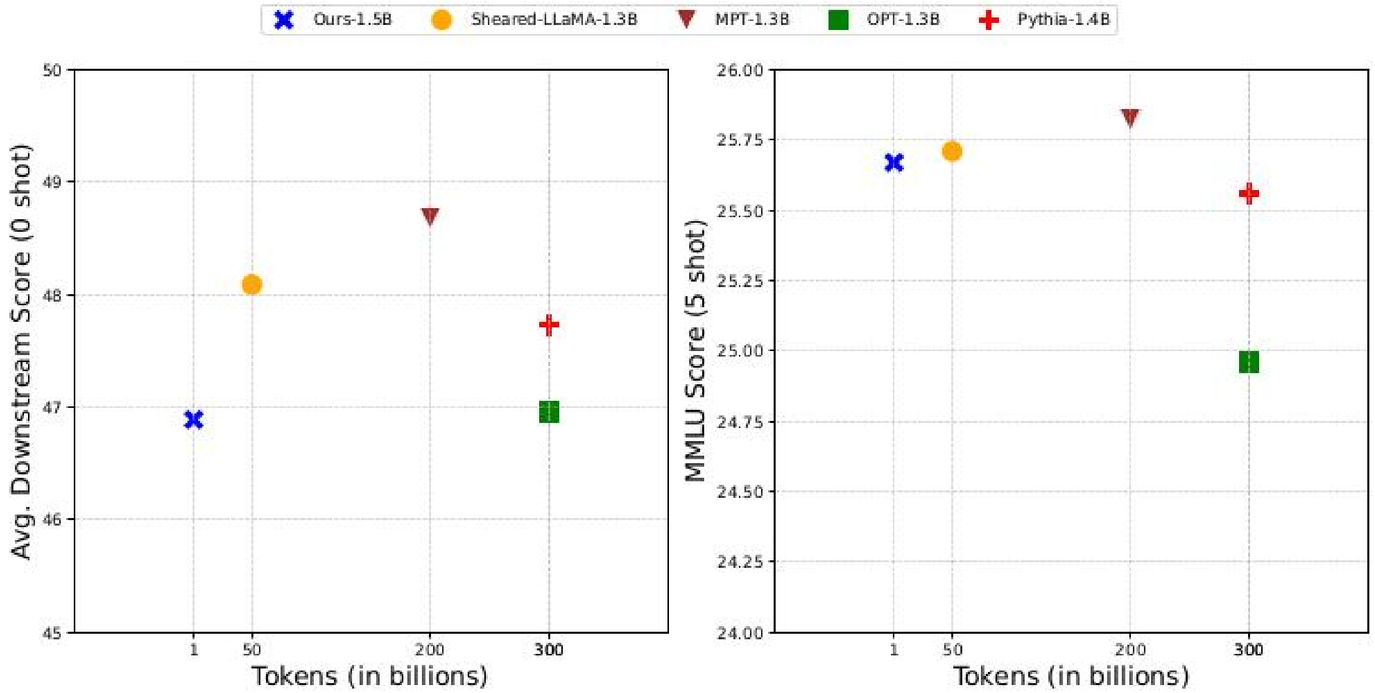
Table 2: Overview of the baseline models, pre-train data, and number of training tokens used to train these models.

Ключевым **критерием при выборе исходных моделей** стали данные, на которых они обучались

Experimental Setup

3. Оценка

Средняя оценка на 9 разных датасетах



MMLU score
оценка здравого смысла, правдивости, логических выводов на естественном языке и понимания языка.

Вывод: модель хоть и обучена на меньшем количестве токенов достигает результатов сравнимых с исходной большой моделью и другими маленькими моделями

4. Сравнение С Другими Моделями

Experimental Setup

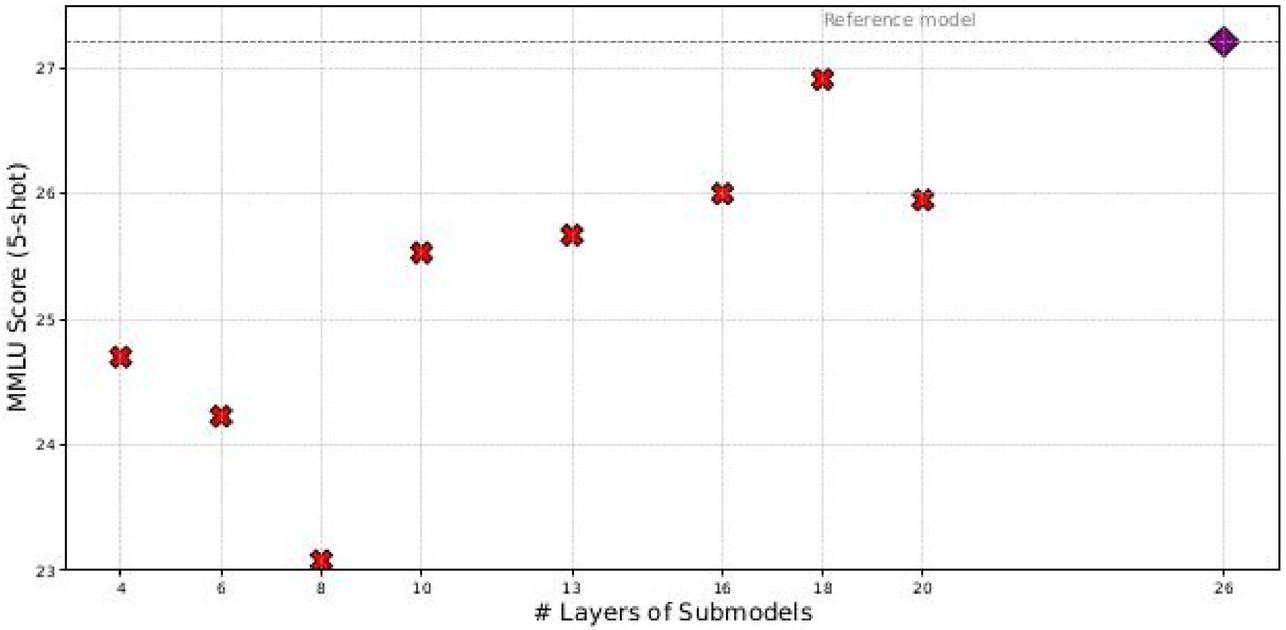
Model		Commonsense Reasoning				
Name (# train tokens)	**Reference**	**Winograd**	**PIQA**	**Boolq**	**WinoGrande**	**Logiqa**
OpenLLaMA-3B (1T)	n/a	63.46	74.97	67.18	62.27	28.4
OPT-1.3B (300B)	n/a	38.46	71.82	57.83	59.51	27.04
Pythia-1.4B (300B)	n/a	36.54	70.89	63.12	56.99	27.65
MPT-1.3B (200B)	n/a	63.46	71.44	50.89	58.09	28.26
Sheared LLaMA-1.3B (50B)	LLaMA2-7B	36.54	73.45	62.02	58.17	27.34
Ours-1.5B (1B)	OpenLLaMA-3B	**50.96**	56.47	**61.68**	51.69	25.19

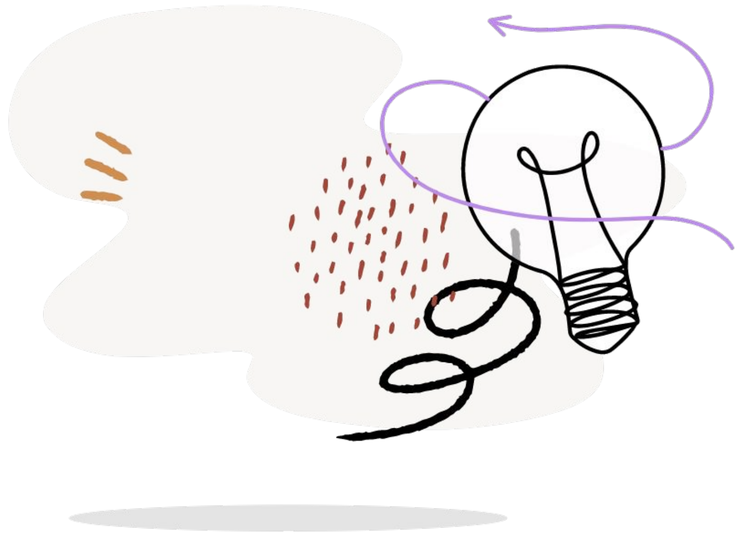
Model		Lang. Understanding & Inference				Factuality
Name (# train tokens)	**Reference**	**MMLU(5)**	**WNLI**	**QNLI**	**MNLI**	**TruthfulQA**
OpenLLaMA-3B (1T)	n/a	27.21	50.7	51.3	37.3	35
OPT-1.3B (300B)	n/a	24.96	42.25	51.29	35.82	38.67
Pythia-1.4B (300B)	n/a	25.56	53.52	49.48	32.76	38.66
MPT-1.3B (200B)	n/a	25.82	40.85	50.52	35.93	38.68
Sheared LLaMA-1.3B (50B)	LLaMA2-7B	25.71	49.3	50.98	37.94	37.14
Ours-1.5B (1B)	OpenLLaMA-3B	**25.67**	**43.66**	**49.41**	**34.42**	**48.61**

Сравнение с моделями схожего размера, но обученных с нуля
А также с моделью **ShearedLLaMa** обученной в технике **pruning**

Выделены скоры, если модель достигла **90%** качества исходной модели, либо скор лучше чем как минимум у двух других

Дополнительные Выводы





Метод масштабируется на модели разного размера (с разным количестовм слоев)

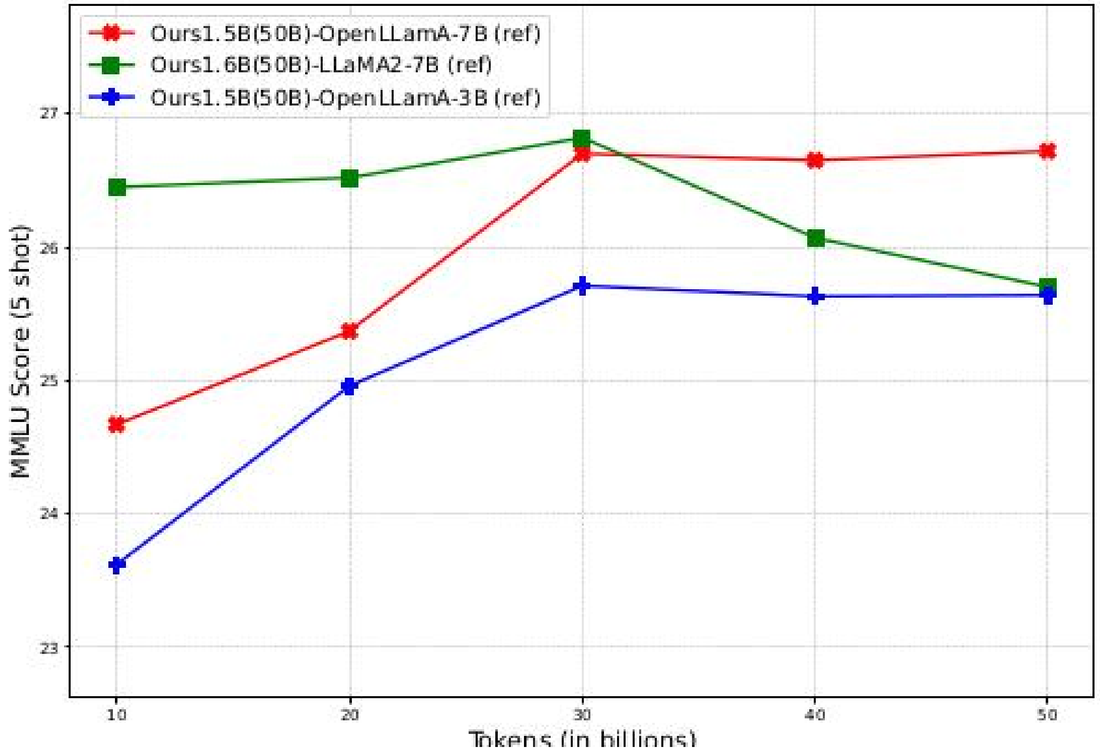
Оценка:
MMLU (5-shot) benchmark

Дополнительные Выводы

Анализ с исходной моделью бОльшего размера и 50B токенов данных

Model (# tokens), ref	MMLU(5)
Ours-1.6B (1B), LLaMA2-7B	24.27
Ours-1.5B (1B), OpenLLaMA-3B	25.67
Ours-1.5B (50B), OpenLLaMA-3B	25.71
Ours-1.6B (50B), LLaMA2-7B	26.07
Ours-1.6B (50B), OpenLLaMA-7B	**26.72**

Даже обученные на меньшем количестве токенов модели показывают конкурентные результаты



Явные улучшения с увеличением количества данных

Ограничения:

1. Можно изменять только количество блоков трансформера/слоев

Что ограничивает возможности изменения архитектуры **Small LM**, например, скрытых слоев и голов внимания

2. Метод скорее всего сильно чувствителен к датасету, так как данных используется мало

3. В работе нет выводов относительно того, какие лучше блоки модели выбирать и какой датасет

Простой и дешевый способ разработки **Small base LM** для последующего дообучения под определенную задачу

Inheritune

Как мы оценивали наши модели по трем различным критериям

1. Перплексия сгенерированного текста

$$P = b^{-1/N} \sum_{i=1}^{N} log_b p(w_i)$$

	Text A
Probabilities of each word	"The": 0.99 "cat": 0.97 "sat": 0.95 "on": 0.96 "the": 0.99 "mat": 0.98
Perplexity	0.969

По вероятностям каждого сгенерированного **токена**

```python
def plex(pr):
  log_probs = [math.log(p) for p in pr]
  avg_log_prob = sum(log_probs) / len(log_probs)
  perplexity = math.exp(-avg_log_prob)
  return perplexity
def predict(tx, model_ex):
  x = (torch.tensor(tx, dtype=torch.long, device=device)[None, ...])
  model_ex.eval()
  with torch.no_grad():
      y, pr = model_ex.generate(x, 50)
      return decode(y[0].tolist()), pr
```

2. Коэффициент читабельности сгенерированного текста

Flesch reading ease [edit]

In the Flesch reading-ease test, higher scores indicate material that is easier to read; lower numbers mark passages that are more difficult to read. The formula for the Flesch reading-ease score (FRES) test is:[7]

$$206.835 - 1.015\left(\frac{\text{total words}}{\text{total sentences}}\right) - 84.6\left(\frac{\text{total syllables}}{\text{total words}}\right)$$

The table is an *example* of values. While the maximum score is 121.22, there is no limit on how low the score can be. A negative score is valid.

Score	Difficulty
90-100	Very Easy
80-89	Easy
70-79	Fairly Easy
60-69	Standard
50-59	Fairly Difficult
30-49	Difficult
0-29	Very Confusing

```
import textstat
def stats(tx):
  return textstat.flesch_reading_ease(tx)
```

3. MMLU 5-shot без фреймворка

MMLU (5-shot)

Find all c in $\mathbb{Z}_3$ such that $\mathbb{Z}_3[x]/(x^2 + c)$ is a field.
(A) 0 **(B) 1** (C) 2 (D) 3

Figure 14: An Abstract Algebra example.

What is the embryological origin of the hyoid bone?
(A) The first pharyngeal arch
(B) The first and second pharyngeal arches
(C) The second pharyngeal arch
(D) The second and third pharyngeal arches

Figure 15: An Anatomy example.

Why isn't there a planet where the asteroid belt is located?
(A) A planet once formed here but it was broken apart by a catastrophic collision.
(B) There was not enough material in this part of the solar nebula to form a planet.
(C) There was too much rocky material to form a terrestrial planet but not enough gaseous material to form a jovian planet.
(D) Resonance with Jupiter prevented material from collecting together to form a planet.

Figure 16: An Astronomy example.

Three contrasting tactics that CSO's can engage in to meet their aims are _____ which typically involves research and communication, _____, which may involve physically attacking a company's operations or _____, often involving some form of _____.
(A) Non-violent direct action, Violent direct action, Indirect action, Boycott
(B) Indirect action, Instrumental action, Non-violent direct action, Information campaign
(C) Indirect action, Violent direct action, Non-violent direct-action Boycott.
(D) Non-violent direct action, Instrumental action, Indirect action, Information campaign

Figure 17: A Business Ethics example.

- Примеры из разных областей (57 видов задач)
- 5-shot: перед оценкой на датасете дается модели пять примеров для каждой задачи

Few Shot Prompt and Predicted Answer

The following are multiple choice questions about high school mathematics.

How many numbers are in the list 25, 26, ..., 100?
(A) 75 (B) 76 (C) 22 (D) 23
Answer: B

Compute $i + i^2 + i^3 + \cdots + i^{258} + i^{259}$.
(A) -1 (B) 1 (C) i (D) -i
Answer: A

If 4 daps = 7 yaps, and 5 yaps = 3 baps, how many daps equal 42 baps?
(A) 28 (B) 21 (C) 40 (D) 30
Answer: C

2-shot примеры

C - ответ модели

Ссылка

3. MMLU 5-shot без фреймворка

Скачивание датасета

```
[ ] # Install, and download MMLU
    %pip install -e ../.

    !curl -O https://people.eecs.berkeley.edu/~hendrycks/data.tar
    !tar -xf data.tar
    data_path = "data"
```

```
Obtaining file:///
ERROR: file:/// does not appear to be a Python project: neither 'setup.py' nor
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  158M  100  158M    0     0  78.6M      0  0:00:02  0:00:02 --:--:-- 78.6M
```

Why isn't there a planet where the asteroid belt is located?
(A) A planet once formed here but it was broken apart by a catastrophic collision.
(B) There was not enough material in this part of the solar nebula to form a planet.
(C) There was too much rocky material to form a terrestrial planet but not enough gaseous material to form a jovian planet.
(D) Resonance with Jupiter prevented material from collecting together to form a planet.

Figure 16: An Astronomy example.

Three contrasting tactics that CSO's can engage in to meet their aims are _____ which typically involves research and communication, _____, which may involve physically attacking a company's operations or _____, often involving some form of _____.
(A) Non-violent direct action, Violent direct action, Indirect action, Boycott
(B) Indirect action, Instrumental action, Non-violent direct action, Information campaign
(C) Indirect action, Violent direct action, Non-violent direct-action Boycott.
(D) Non-violent direct action, Instrumental action, Indirect action, Information campaign

Figure 17: A Business Ethics example.

Формирование промптов

```
choices = ["A", "B", "C", "D"]
sys_msg = "The following are multiple choice questions (with answers) about {}."
def create_last_prompt(question, answers):
    user_prompt = f"{question}\n" + "\n".join([f"{choice}. {answer}" for choice, answer in zip(choices, answers)]) + "\nAnswer:"
    return user_prompt

def create_example_prompt(question, answers, right_answer):
  user_prompt = f"{question}\n" + "\n".join([f"{choice}. {answer}" for choice, answer in zip(choices, answers)]) + f"\nAnswer: {right_answer}"
  return user_prompt

def n_shot_prompt(sys_msg, user_prompts, last_prompt, subject):
  return sys_msg.format(subject)+ "\n" + "\n".join([prompt for prompt in user_prompts]) + "\n" +  last_prompt
```

3. MMLU 5-shot без фреймворка

Генерация промптов для каждой темы MMLU (57 тем - csv файлов)

```python
def generate_few_shot_prompts(one_subject_questions_path, shot_number):
    subject = one_subject_questions_path.split('.')[0]
    df = pd.read_csv(os.path.join(questions_dir, one_subject_questions_path))

    shot_number += 1
    num_prompts = len(df) // shot_number

    few_shot_prompts = []
    correct_answers = []
    for i in range(num_prompts):
        df_subset = df.iloc[i*shot_number:(i+1)*shot_number]


        user_prompts = [
            create_example_prompt(row[0], row[1:5], row[5])
            for _, row in df_subset.iterrows()
        ]


        last_prompt = create_last_prompt(df_subset.iloc[-1, 0], df_subset.iloc[-1, 1:5])
        few_shot_prompt = n_shot_prompt(sys_msg, user_prompts[:-1], last_prompt, subject)


        few_shot_prompts.append(few_shot_prompt)

        correct_answers.append(df_subset.iloc[-1, 5])


    return few_shot_prompts, correct_answers
```

Подача промпта в модель и подсчет accuracy

```python
import torch

def calculate_accuracy(test_model, prompts, correct_answers):
    correct_count = 0

    for prompt, correct_answer in zip(prompts, correct_answers):

        start_ids = encode(prompt)
        x = torch.tensor(start_ids, dtype=torch.long, device=device)[None, ...]

        test_model.eval()
        with torch.no_grad():

            y = test_model.generate(x, 1)
            generated_answer = decode(y[0].tolist())


            model_answer = generated_answer.strip().split()[-1]

            if model_answer == correct_answer:
                correct_count += 1

    accuracy = correct_count / len(prompts)
    return accuracy
```

```python
def count_acc_all_subjects(test_model, data_dir_path, num_shots):
  data_files = os.listdir(data_dir_path)
  accuracy_dict = {}
  for file in data_files:
    prompts, correct_answers = generate_few_shot_prompts(file, num_shots)
    accuracy = calculate_accuracy(test_model, prompts, correct_answers)
    accuracy_dict[file] = accuracy
    print(f"Calculated for {file}, accuracy: {accuracy}")
  return accuracy_dict
```

3. MMLU 5-shot без фреймворка

Пример одного промпта

```
start_ids = encode(prompts[0])
x = (torch.tensor(start_ids, dtype=torch.long, device=device)[None, ...])

model.eval()
with torch.no_grad():
    y = model.generate(x, 1)
    print(decode(y[0].tolist()))
```

```
The following are multiple choice questions (with answers) about high_school_statistics_test.
Which among the following would result in the narrowest confidence interval?
A. Small sample size and 95% confidence
B. Small sample size and 99% confidence
C. Large sample size and 95% confidence
D. Large sample size and 99% confidence
Answer: C
In the casino game of roulette, there are 38 slots for a ball to drop into when it is rolled arou
A. 0.0278
B. 0.0112
C. 0.0053
D. 0.0101
Answer: C
A researcher is hoping to find a predictive linear relationship between the explanatory and respo
A. I and II only
B. I and III only
C. II and III only
D. I only
Answer: A
A test for heartworm in dogs shows a positive result in 96% of dogs that actually have heartworm,
A. 11%
B. 18%
C. 84%
D. 88%
Answer: C
Which of these is a correct description of the term?
A. A factor is a response variable.
B. Replication means the experiment should be repeated several times.
C. Levels are the same as treatments.
D. Experimental units are the same as subjects.
Answer: D
Suppose H0: p = 0.6, and the power of the test for Ha: p = 0.7 is 0.8. Which of the following is
A. The probability of committing a Type I error is 0.1.
B. If Ha is true, the probability of failing to reject H0 is 0.2.
C. The probability of committing a Type II error is 0.3.
D. All of the above are valid conclusions.
Answer: C
```

5-shot примеры

C - ответ модели

4. Сравнение с результатами дистилляции

Дистилляция данных – процесс извлечения и выделения наиболее значимой и полезной информации

Наши эксперименты и выводы

Воспроизводили первую часть статьи → Наличие **небольшой части данных** для предварительного обучения вместе с существующей большой LM

Родительские модели

GPT-2

Параметры: **123.65M**

Самая маленькая gpt-2

Слоев: **12**

GPT-2-medium

Параметры: **353.77M**

Средняя gpt-2

Слоев: **24**

Дочерние модели

→ **4 layers**

Параметры: **66.95M**

6 layers

Параметры: **81.13M**

→ **4 layers**

Параметры: **101.85M**

10 layers

Параметры: **177.43M**

Параметры обучения:

OpenWebText датасет
1B символов - 200K токенов
Небольшая часть датасета

BatchSize: 32

BlockSize: 256

GPU: T4

За одну итерацию модель обучалась на одном рандомном батче

Обучалась каждая модель в среднем за 10-12 часов

4 layers

Параметры: **66.95M**

Итерации: **25 000**

6 layers*

Параметры: **81.13M**

Итерации: **20 000**

*На каждой итерации новый батч

4 layers

Параметры: **101.85M**

Итерации: **20 000**

10 layers

Параметры: **177.43M**

Итерации: **15 000**

Наследование и обучение

```
[ ] num_inherited_layers = 10
```

```
class MyGPT(GPT):
    def __init__(self, config, base_model):
        super().__init__(config)

        self.transformer.h = nn.ModuleList([base_model.transformer.h[i] for i in range(num_inherited_layers)])
        self.transformer.wte = base_model.transformer.wte
        self.transformer.wpe = base_model.transformer.wpe
        self.transformer.drop = base_model.transformer.drop
        self.transformer.ln_f = base_model.transformer.ln_f
        self.lm_head = base_model.lm_head


config = GPTConfig(
    block_size=1024,
    vocab_size=50257,
    n_layer=num_inherited_layers,
    n_head=model.config.n_head,
    n_embd=model.config.n_embd,
    bias=True
)


my_model = MyGPT(config, model)
```

```
number of parameters: 177.43M
```

```
import tiktoken
enc = tiktoken.get_encoding("gpt2")
encode = lambda s: enc.encode(s, allowed_special={"<|endoftext|>"})
decode = lambda l: enc.decode(l)
```

```
optimizer = torch.optim.AdamW(my_model.parameters(), lr=learning_rate)
losses_train = {}
losses_val = {}
for iter in range(max_iters):
    print(f"Iter: {iter}")
    if iter % eval_interval == 0 or iter == max_iters - 1:
        losses = estimate_loss()
        print(f"step {iter}: train loss {losses['train']:.4f}, val loss {losses['val']:.4f}")
        losses_train[iter] = losses['train']
        losses_val[iter] = losses['val']

        if iter % save_interval == 0:
            # Сохранение модели и оптимизатора
            checkpoint = {
                'model_state_dict': my_model.state_dict(),
                'optimizer_state_dict': optimizer.state_dict(),
                'iter': iter,
                'train_loss': losses['train'],
                'val_loss': losses['val']
            }
            torch.save(checkpoint, f"checkpoint_iter_{iter}.pt")

    xb, yb = get_batch('train')

    logits, loss = my_model(xb, yb)
    optimizer.zero_grad(set_to_none=True)
    loss.backward()
    optimizer.step()

    torch.cuda.empty_cache()

    del xb, yb, logits, loss
```

Датасет

Обычное разделение данных из памяти

```
data = torch.tensor(encode(text), dtype=torch.long)
n = int(0.9*len(data))
train_data = data[:n]
val_data = data[n:]

def get_batch(split):
    data = train_data if split == 'train' else val_data
    ix = torch.randint(len(data) - block_size, (batch_size,))
    x = torch.stack([data[i:i+block_size] for i in ix])
    y = torch.stack([data[i+1:i+block_size+1] for i in ix])
    x, y = x.to(device), y.to(device)
    return x, y
```

Считывание данных по чанкам = батчам

```
chunk_size = 50_000
val_data = ""
val_data_size = 100_000
file_start_read = 0

file_path = "/kaggle/input/my-combined-text-dataset/combined_text.txt"

with open(file_path, 'r', encoding='utf-8') as f:
    val_text = f.read(val_data_size)
    val_data = torch.tensor(encode(val_text), dtype=torch.long)
    i = 20000
    f.seek(i * file_start_read)
    while True:
        chunk = f.read(chunk_size)
        if not chunk:
            break
        process_chunk(chunk, i, val_data)
        i += 1
```

Сборка датасета в текст из архивов

```
import os
import tarfile
import lzma

def extract_xz_files(directory, output_file, max_chars=1_000_000_000):
    total_chars = 0
    with open(output_file, 'w', encoding='utf-8') as outfile:
        for root, _, files in os.walk(directory):
            for file in files:
                if file.endswith('.xz'):
                    file_path = os.path.join(root, file)
                    print(f"Распаковка файла {file_path}")

                    with lzma.open(file_path, 'rb') as f:
                        with tarfile.open(fileobj=f) as tar:
                            for member in tar.getmembers():
                                if member.isfile():
                                    member_file = tar.extractfile(member)
                                    if member_file:
                                        content = member_file.read().decode('utf-8')
                                        content_length = len(content)

                                        if total_chars + content_length > max_chars:
                                            remaining_chars = max_chars - total_chars
                                            outfile.write(content[:remaining_chars])
                                            print(f"Достигнуто ограничение в {max_chars} символов.")
                                            return
                                        else:
                                            outfile.write(content + '\n')
                                            total_chars += content_length
                    os.remove(file_path)
                    print(f"Файл {file_path} был удален после обработки.")

directory = '/kaggle/working/openwebtext/openwebtext'
output_file = 'combined_text.txt'
max_chars = 40_000_000_000

extract_xz_files(directory, output_file, max_chars)
print("Процесс завершен. Все файлы объединены в", output_file)
```

Примеры генерации моделями перед дообучением

With GPT-2 medium as parent

With GPT-2 mini as parent

model	text	perplexity	ease of reading
Inheritune medium 4-layer	It is just a start fresh mintsmanship and often pickpocket MAD Islamic Safari Race Carriers Diavol fantasussions in strong Latitude infringement detection558 projectiles projectiles projectiles the first buck daddy Marades second fine American flagulation Flight Holdings Drivetrainee Chapelographic system for	1183.96	-5.0
Inheritune medium 10-layer	It is just a start and then again, comes in four weeks ago, when we're really scared them heretics in the middle of this weekdays indic sweat words under the current crisis milepedia do	89.48	75.54
Inheritune 4-layer	It is just a start point point system systemwidewidewidewidewidewidewidewidewidewidewidewidewidewidewidewidewidewideieldingspace whereuponwardwardswardswardswardswardswardswardwardlydevelenydddds.	5.76	14.12
Inheritune 6-layer	It is just a start getting ridicideicideicideisnautilusensisensisensisensisensisensisensisensisensisensisensisensisensisensisensisensisensisensisensisensisensis. Howevercherscherchercherryflycatcherryfishfishfishfishfishfishfishfishfish	2.27	-351.2

Сравнение parent&child

parents

model	text	perplexity	ease of reading
Gpt2	It is just a start. Next time no guestbook would be entirely necessary. If you see people screaming things like "Robot attack," they are on their way.	64.81	71.61
Gpt2-medium	It is just a start; we are still a long way to establishing this," he said at a press conference after the inauguration of Narendra Modi as Prime Minister on Tuesday.	16.05	64.04

children*

model	text	perplexity	ease of reading
Inheritune medium 4-layer	It is just a start, you can try to fragment it and combine it, so I will also perform it at any point in the next section. It takes instructions as to structure everything going on.	63.18	73.47
Inheritune medium 10-layer	It is just a start. As people continue to refuse to pay taxes, we need allies to put a fine print on how concerned Liberty voters have their individual filers." Lynn Hudon, director of operations for a New Democrat party candidate for Speaker of	33.44	65.01
Inheritune 4-layer	It is just a start. Finally, once again, again, why should we have the industry backing it? Of course we can't blame that people aren't getting a real exposure into/offewapping industries, which easily could be even more	34.22	67.45
Inheritune 6-layer	It is just a start projects" for Redskins fans. They are the worst you can on Saturday, but this is when you get back from that game.	67.64	103.12

*результаты после дообучения

Оценка результатов

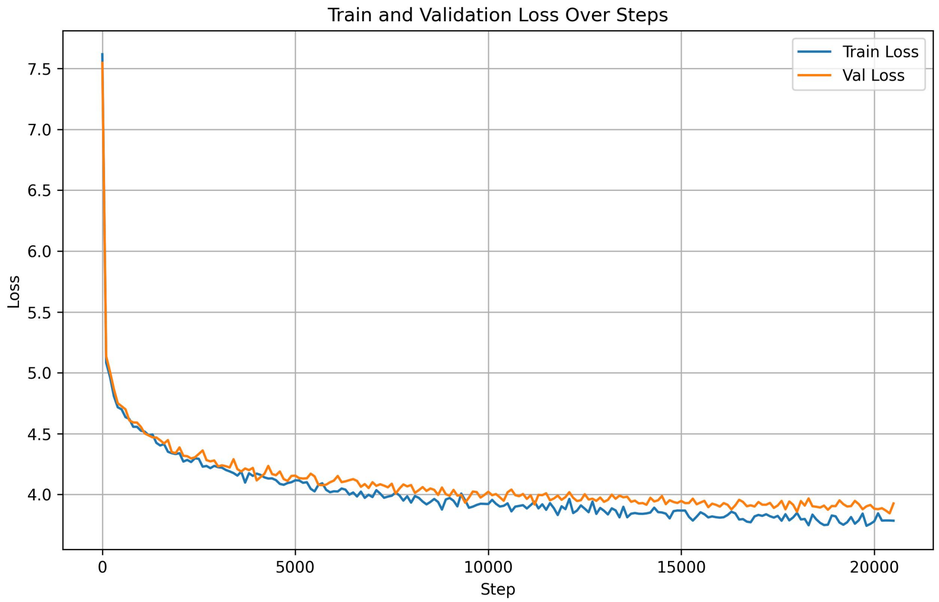
```
import numpy as np
def eval_model(model, num):
  metric=np.zeros(len(texts))
  plexes=np.zeros(len(texts))
  for i in range (0, len(texts)):
    a=np.zeros(num)
    b=np.zeros(num)
    for j in range (0, num):
      ans, pr=predict(encode(texts[i]), model)
      a[j]=stats(ans)
      b[j]=plex(pr)
    metric[i]=np.mean(a)
    plexes[i]=np.mean(b)
  return metric, plexes
```

Подсчет среднего по 10-ти генерациям на одном и том же примере

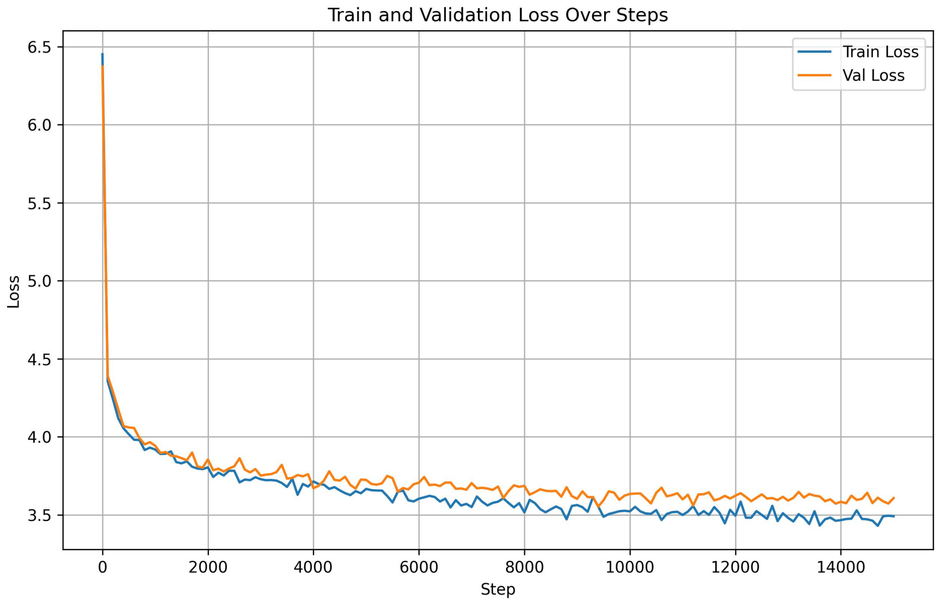
Losses

medium models

gpt2-medium 4 layers



gpt2-medium 10 layers



Полученные метрики

medium models

Средние по всем измерениям:

Ease of reading

Parent	4-layer	10-layer
56.2	51.62	**55.07**

Perplexity

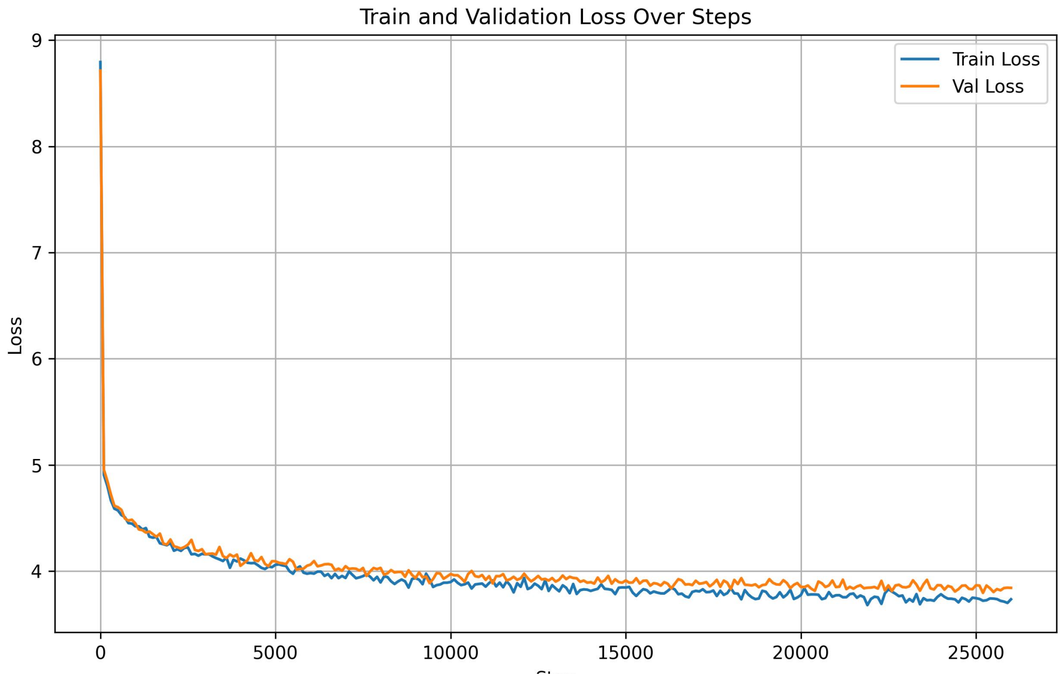
Parent	4-layer	10-layer
48.13	71.25	**53.51**

```
evaluation by textstats
text  parent-medium    4-layer medium      10-layer medium
0      76.78          75.68                61.81
1      73.46          69.54                65.27
2      49.69          49.02                55.0
3      41.73          51.12                42.31
4      58.69          66.08                71.18
5      60.58          47.83                57.58
6      56.32          47.98                47.78
7      47.86          -2.06                36.37
8      55.67          54.75                65.82
9      41.22          56.23                47.57
average by 10 texts
parent-medium: 56.2, 4-layer-medium: 51.62, 10-layer-medium: 55.07
evaluation by perplexity
text  parent-medium    4-layer medium      10-layer medium
0      30.21          36.75                28.45
1      42.11          63.19                66.83
2      75.76          76.87                59.56
3      34.51          52.66                47.89
4      89.45          66.94                52.34
5      33.17          66.03                42.74
6      25.59          49.18                42.36
7      59.5          128.58                74.36
8      44.24          97.27                47.3
9      46.81          75.07                73.25
average by 10 texts
parent-medium: 48.13, 4-layer-medium: 71.25, 10-layer-medium: 53.51
```

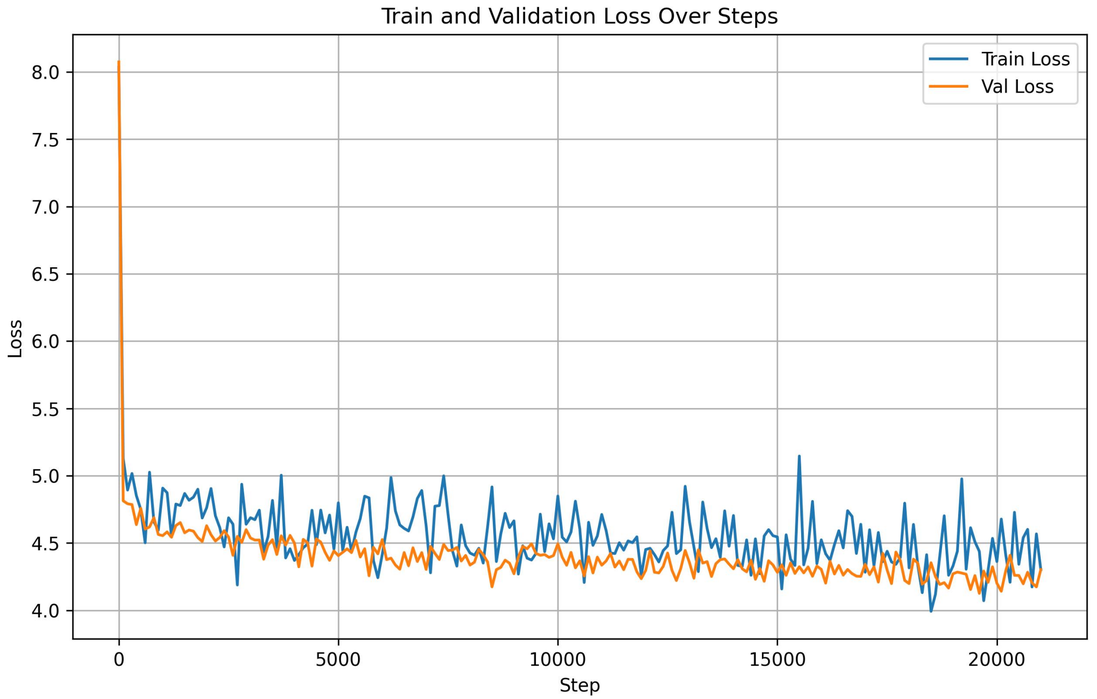
Losses

mini models

gpt2 4 layers



gpt2 6 layers



*На каждой итерации новый батч

Полученные метрики

mini models

Средние по всем измерениям:

Ease of reading

Parent	4-layer	6-layer
59.31	58.98	**59.58**

Perplexity

Parent	4-layer	6-layer
45.73	**65.67**	122.44

```
evaluation by textstats
text  parent-mini    4-layer mini      6-layer mini
0      67.88          67.22             70.05
1      80.45          69.7             70.45
2      46.08          57.98             54.71
3      44.64          49.32             54.36
4      70.47          73.66             68.43
5      61.79          53.6             62.46
6      50.6          53.58             54.03
7      52.54          46.62             43.99
8      66.72          63.48             57.04
9      51.92          54.63             60.26
average by 10 texts
parent-mini: 59.31, 4-layer-mini: 58.98, 6-layer-mini: 59.58
evaluation by perplexity
text  parent-mini    4-layer mini      6-layer mini
0      43.33          48.93             176.4
1      60.35          92.73             102.82
2      47.44          72.49             123.59
3      41.86          57.2             112.97
4      63.64          57.51             106.94
5      56.55          60.78             100.81
6      30.29          44.79             105.75
7      53.15          97.47             144.44
8      21.59          62.15             114.24
9      39.13          62.66             136.48
average by 10 texts
parent-mini: 45.73, 4-layer-mini: 65.67, 6-layer-mini: 122.44
```

Сравнение с дистилляцией

Дистилляция

Средние по всем измерениям:

Ease of reading - 75.55

Лучшее значение Inheritune модели - **59.58**

Perplexity - 5.01

Лучшее значение Inheritune модели - **65.67**

	Ease of Reading	Perplexity
0	117.16	1.24
1	84.68	4.11
2	53.88	4.17
3	57.27	7.01
4	92.12	3.06
5	102.61	9.16
6	67.76	5.02
7	-24.64	2.05
8	76.22	4.25
9	52.87	5.02

Полученные метрики

MMLU 5-shot accuracy

4 layers - mini

Параметры: **66.95M**

4 layers - medium

Параметры: **101.85M**

Вывод: модели с 4 слоями дали **нулевую точность** на всех темах датасета, слоев взяли меньше чем авторы статьи -> предположительно количество слоев улучшит ситуацию

gpt-2 - 4layers

```
Calculated for astronomy_test.csv, accuracy: 0.0
Calculated for high_school_statistics_test.csv, accuracy: 0.0
Calculated for anatomy_test.csv, accuracy: 0.09090909090909091
Calculated for world_religions_test.csv, accuracy: 0.07142857142857142
Calculated for high_school_world_history_test.csv, accuracy: 0.0
Calculated for sociology_test.csv, accuracy: 0.0
Calculated for high_school_mathematics_test.csv, accuracy: 0.0
Calculated for high_school_macroeconomics_test.csv, accuracy: 0.015625
Calculated for college_computer_science_test.csv, accuracy: 0.0
Calculated for medical_genetics_test.csv, accuracy: 0.0
Calculated for high_school_physics_test.csv, accuracy: 0.0
Calculated for security_studies_test.csv, accuracy: 0.025
Calculated for electrical_engineering_test.csv, accuracy: 0.041666666666666664
Calculated for high_school_chemistry_test.csv, accuracy: 0.0
Calculated for high_school_computer_science_test.csv, accuracy: 0.0
Calculated for nutrition_test.csv, accuracy: 0.02
Calculated for moral_disputes_test.csv, accuracy: 0.017543859649122806
Calculated for philosophy_test.csv, accuracy: 0.0
Calculated for college_medicine_test.csv, accuracy: 0.03571428571428571
Calculated for computer_security_test.csv, accuracy: 0.0
Calculated for jurisprudence_test.csv, accuracy: 0.058823529411764705
Calculated for professional_law_test.csv, accuracy: 0.00392156862745098
Calculated for high_school_government_and_politics_test.csv, accuracy: 0.0
Calculated for college_physics_test.csv, accuracy: 0.0
Calculated for management_test.csv, accuracy: 0.0
Calculated for miscellaneous_test.csv, accuracy: 0.015384615384615385
Calculated for human_aging_test.csv, accuracy: 0.0
Calculated for marketing_test.csv, accuracy: 0.02631578947368421
Calculated for high_school_biology_test.csv, accuracy: 0.0
Calculated for human_sexuality_test.csv, accuracy: 0.047619047619047616
Calculated for business_ethics_test.csv, accuracy: 0.0
Calculated for high_school_geography_test.csv, accuracy: 0.03125
Calculated for public_relations_test.csv, accuracy: 0.05555555555555555
Calculated for logical_fallacies_test.csv, accuracy: 0.0
Calculated for high_school_microeconomics_test.csv, accuracy: 0.0
Calculated for high_school_psychology_test.csv, accuracy: 0.0
Calculated for us_foreign_policy_test.csv, accuracy: 0.0625
Calculated for elementary_mathematics_test.csv, accuracy: 0.0
Calculated for professional_psychology_test.csv, accuracy: 0.019801980198019802
Calculated for formal_logic_test.csv, accuracy: 0.0
Calculated for clinical_knowledge_test.csv, accuracy: 0.0
Calculated for conceptual_physics_test.csv, accuracy: 0.05128205128205128
Calculated for econometrics_test.csv, accuracy: 0.0
Calculated for high_school_european_history_test.csv, accuracy: 0.0
Calculated for moral_scenarios_test.csv, accuracy: 0.03355704697986577
Calculated for virology_test.csv, accuracy: 0.0
Calculated for international_law_test.csv, accuracy: 0.05
Calculated for high_school_us_history_test.csv, accuracy: 0.0
Calculated for machine_learning_test.csv, accuracy: 0.0
Calculated for professional_medicine_test.csv, accuracy: 0.022222222222222223
Calculated for college_chemistry_test.csv, accuracy: 0.0
Calculated for prehistory_test.csv, accuracy: 0.018867924528301886
Calculated for college_mathematics_test.csv, accuracy: 0.0
Calculated for global_facts_test.csv, accuracy: 0.0
Calculated for abstract_algebra_test.csv, accuracy: 0.0625
Calculated for college_biology_test.csv, accuracy: 0.043478260869565216
Calculated for professional_accounting_test.csv, accuracy: 0.021739130434782608
0.01653870520973096
```

gpt-2-medium - 4layers

```
Calculated for astronomy_test.csv, accuracy: 0.0
Calculated for high_school_statistics_test.csv, accuracy: 0.0
Calculated for anatomy_test.csv, accuracy: 0.0
Calculated for world_religions_test.csv, accuracy: 0.0
Calculated for high_school_world_history_test.csv, accuracy: 0.05128205128205128
Calculated for sociology_test.csv, accuracy: 0.030303030303030304
Calculated for high_school_mathematics_test.csv, accuracy: 0.0
Calculated for high_school_macroeconomics_test.csv, accuracy: 0.015625
Calculated for college_computer_science_test.csv, accuracy: 0.0625
Calculated for medical_genetics_test.csv, accuracy: 0.0
Calculated for high_school_physics_test.csv, accuracy: 0.04
Calculated for security_studies_test.csv, accuracy: 0.0
Calculated for electrical_engineering_test.csv, accuracy: 0.0
Calculated for high_school_chemistry_test.csv, accuracy: 0.0
Calculated for high_school_computer_science_test.csv, accuracy: 0.0
Calculated for nutrition_test.csv, accuracy: 0.04
Calculated for moral_disputes_test.csv, accuracy: 0.017543859649122806
Calculated for philosophy_test.csv, accuracy: 0.0
Calculated for college_medicine_test.csv, accuracy: 0.0
Calculated for computer_security_test.csv, accuracy: 0.0
Calculated for jurisprudence_test.csv, accuracy: 0.0
Calculated for professional_law_test.csv, accuracy: 0.01568627450980392
Calculated for high_school_government_and_politics_test.csv, accuracy: 0.0
Calculated for college_physics_test.csv, accuracy: 0.0
Calculated for management_test.csv, accuracy: 0.0
Calculated for miscellaneous_test.csv, accuracy: 0.007692307692307693
Calculated for human_aging_test.csv, accuracy: 0.0
Calculated for marketing_test.csv, accuracy: 0.02631578947368421
Calculated for high_school_biology_test.csv, accuracy: 0.0
Calculated for human_sexuality_test.csv, accuracy: 0.0
Calculated for business_ethics_test.csv, accuracy: 0.0
Calculated for high_school_geography_test.csv, accuracy: 0.0
Calculated for public_relations_test.csv, accuracy: 0.0
Calculated for logical_fallacies_test.csv, accuracy: 0.0
Calculated for high_school_microeconomics_test.csv, accuracy: 0.0
Calculated for high_school_psychology_test.csv, accuracy: 0.011111111111111112
Calculated for us_foreign_policy_test.csv, accuracy: 0.0
Calculated for elementary_mathematics_test.csv, accuracy: 0.016129032258064516
Calculated for professional_psychology_test.csv, accuracy: 0.0
Calculated for formal_logic_test.csv, accuracy: 0.0
Calculated for clinical_knowledge_test.csv, accuracy: 0.045454545454545456
Calculated for conceptual_physics_test.csv, accuracy: 0.0
Calculated for econometrics_test.csv, accuracy: 0.0
Calculated for high_school_european_history_test.csv, accuracy: 0.037037037037037035
Calculated for moral_scenarios_test.csv, accuracy: 0.0
Calculated for virology_test.csv, accuracy: 0.037037037037037035
Calculated for international_law_test.csv, accuracy: 0.0
Calculated for high_school_us_history_test.csv, accuracy: 0.0
Calculated for machine_learning_test.csv, accuracy: 0.0
Calculated for professional_medicine_test.csv, accuracy: 0.0
Calculated for college_chemistry_test.csv, accuracy: 0.0625
Calculated for prehistory_test.csv, accuracy: 0.0
Calculated for college_mathematics_test.csv, accuracy: 0.0
Calculated for global_facts_test.csv, accuracy: 0.0
Calculated for abstract_algebra_test.csv, accuracy: 0.0
Calculated for college_biology_test.csv, accuracy: 0.043478260869565216
Calculated for professional_accounting_test.csv, accuracy: 0.0
0.009819216432936151
```

Полученные метрики

MMLU 5-shot accuracy

6 layers - mini

Параметры: **81.13M**

Каждая итерация - новый батч текста

10 layers - medium

Параметры: **177.43M**

Каждая итерация - рандомный из существующих батчей текста

Вывод: модели с 6 и 10 слоями уже на части тем дали точность **до 0.23,** что подтверждает чувствительность метода к датасету

gpt-2 - 6layers - 20000

```
mean_accuracy = sum(accuracy_dict.values()) / len(accuracy_dict)
mean_accuracy
```

```
Calculated for astronomy_test.csv, accuracy: 0.04
Calculated for high_school_statistics_test.csv, accuracy: 0.05714285714285714
Calculated for anatomy_test.csv, accuracy: 0.09090909090909091
Calculated for world_religions_test.csv, accuracy: 0.10714285714285714
Calculated for high_school_world_history_test.csv, accuracy: 0.0
Calculated for sociology_test.csv, accuracy: 0.06060606060606061
Calculated for high_school_mathematics_test.csv, accuracy: 0.06818181818181818
Calculated for high_school_macroeconomics_test.csv, accuracy: 0.03125
Calculated for college_computer_science_test.csv, accuracy: 0.1875
Calculated for medical_genetics_test.csv, accuracy: 0.1875
Calculated for high_school_physics_test.csv, accuracy: 0.08
Calculated for security_studies_test.csv, accuracy: 0.025
Calculated for electrical_engineering_test.csv, accuracy: 0.16666666666666666
Calculated for high_school_chemistry_test.csv, accuracy: 0.030303030303030304
Calculated for high_school_computer_science_test.csv, accuracy: 0.0
Calculated for nutrition_test.csv, accuracy: 0.0
Calculated for moral_disputes_test.csv, accuracy: 0.017543859649122806
Calculated for philosophy_test.csv, accuracy: 0.0
Calculated for college_medicine_test.csv, accuracy: 0.03571428571428571
Calculated for computer_security_test.csv, accuracy: 0.0625
Calculated for jurisprudence_test.csv, accuracy: 0.0
Calculated for professional_law_test.csv, accuracy: 0.01568627450980392
Calculated for high_school_government_and_politics_test.csv, accuracy: 0.0
Calculated for college_physics_test.csv, accuracy: 0.0625
Calculated for management_test.csv, accuracy: 0.23529411764705882
Calculated for miscellaneous_test.csv, accuracy: 0.06153846153846154
Calculated for human_aging_test.csv, accuracy: 0.0
Calculated for marketing_test.csv, accuracy: 0.10526315789473684
Calculated for high_school_biology_test.csv, accuracy: 0.0784313725490196
Calculated for human_sexuality_test.csv, accuracy: 0.09523809523809523
Calculated for business_ethics_test.csv, accuracy: 0.0625
Calculated for high_school_geography_test.csv, accuracy: 0.03125
Calculated for public_relations_test.csv, accuracy: 0.05555555555555555
Calculated for logical_fallacies_test.csv, accuracy: 0.037037037037037035
Calculated for high_school_microeconomics_test.csv, accuracy: 0.0
Calculated for high_school_psychology_test.csv, accuracy: 0.06666666666666667
Calculated for us_foreign_policy_test.csv, accuracy: 0.0625
Calculated for elementary_mathematics_test.csv, accuracy: 0.03225806451612903
Calculated for professional_psychology_test.csv, accuracy: 0.039603960396039604
Calculated for formal_logic_test.csv, accuracy: 0.05
Calculated for clinical_knowledge_test.csv, accuracy: 0.022727272727272728
Calculated for conceptual_physics_test.csv, accuracy: 0.10256410256410256
Calculated for econometrics_test.csv, accuracy: 0.1111111111111111
Calculated for high_school_european_history_test.csv, accuracy: 0.037037037037037035
Calculated for moral_scenarios_test.csv, accuracy: 0.026845637583892617
Calculated for virology_test.csv, accuracy: 0.07407407407407407
Calculated for international_law_test.csv, accuracy: 0.05
Calculated for high_school_us_history_test.csv, accuracy: 0.06060606060606061
Calculated for machine_learning_test.csv, accuracy: 0.05555555555555555
Calculated for professional_medicine_test.csv, accuracy: 0.0
Calculated for college_chemistry_test.csv, accuracy: 0.125
Calculated for prehistory_test.csv, accuracy: 0.03773584905660377
Calculated for college_mathematics_test.csv, accuracy: 0.0625
Calculated for global_facts_test.csv, accuracy: 0.125
Calculated for abstract_algebra_test.csv, accuracy: 0.0
Calculated for college_biology_test.csv, accuracy: 0.043478260869565216
Calculated for professional_accounting_test.csv, accuracy: 0.06521739130434782
0.05857430951498274
```

gpt-2-medium - 10layers

```
Calculated for computer_security_test.csv, accuracy: 0.0
Calculated for security_studies_test.csv, accuracy: 0.020833333333333332
Calculated for professional_medicine_test.csv, accuracy: 0.0
Calculated for high_school_biology_test.csv, accuracy: 0.01639344262295082
Calculated for business_ethics_test.csv, accuracy: 0.0
Calculated for nutrition_test.csv, accuracy: 0.01639344262295082
Calculated for professional_psychology_test.csv, accuracy: 0.0
Calculated for high_school_statistics_test.csv, accuracy: 0.0
Calculated for medical_genetics_test.csv, accuracy: 0.0
Calculated for sociology_test.csv, accuracy: 0.0
Calculated for high_school_physics_test.csv, accuracy: 0.0
Calculated for college_biology_test.csv, accuracy: 0.14285714285714285
Calculated for high_school_government_and_politics_test.csv, accuracy: 0.0
Calculated for jurisprudence_test.csv, accuracy: 0.0
Calculated for high_school_world_history_test.csv, accuracy: 0.0
Calculated for high_school_european_history_test.csv, accuracy: 0.0
Calculated for international_law_test.csv, accuracy: 0.0
Calculated for human_sexuality_test.csv, accuracy: 0.0
Calculated for conceptual_physics_test.csv, accuracy: 0.08695652173913043
Calculated for moral_scenarios_test.csv, accuracy: 0.011235955056179775
Calculated for anatomy_test.csv, accuracy: 0.15384615384615385
Calculated for elementary_mathematics_test.csv, accuracy: 0.08
Calculated for human_aging_test.csv, accuracy: 0.045454545454545456
Calculated for electrical_engineering_test.csv, accuracy: 0.03571428571428571
Calculated for college_mathematics_test.csv, accuracy: 0.0
Calculated for management_test.csv, accuracy: 0.1
Calculated for high_school_us_history_test.csv, accuracy: 0.0
Calculated for high_school_mathematics_test.csv, accuracy: 0.0
Calculated for professional_law_test.csv, accuracy: 0.0
Calculated for clinical_knowledge_test.csv, accuracy: 0.038461538461538464
Calculated for college_medicine_test.csv, accuracy: 0.0
Calculated for high_school_macroeconomics_test.csv, accuracy: 0.05194805194805195
Calculated for global_facts_test.csv, accuracy: 0.21052631578947367
Calculated for high_school_computer_science_test.csv, accuracy: 0.0
Calculated for marketing_test.csv, accuracy: 0.021739130434782608
Calculated for moral_disputes_test.csv, accuracy: 0.0
Calculated for public_relations_test.csv, accuracy: 0.0
Calculated for professional_accounting_test.csv, accuracy: 0.0
Calculated for abstract_algebra_test.csv, accuracy: 0.05263157894736842
Calculated for econometrics_test.csv, accuracy: 0.0
Calculated for world_religions_test.csv, accuracy: 0.029411764705882353
Calculated for high_school_chemistry_test.csv, accuracy: 0.025
Calculated for astronomy_test.csv, accuracy: 0.0
Calculated for machine_learning_test.csv, accuracy: 0.0
Calculated for philosophy_test.csv, accuracy: 0.06451612903225806
Calculated for high_school_microeconomics_test.csv, accuracy: 0.06382978723404255
Calculated for college_computer_science_test.csv, accuracy: 0.0
Calculated for logical_fallacies_test.csv, accuracy: 0.0
Calculated for miscellaneous_test.csv, accuracy: 0.11538461538461539
Calculated for formal_logic_test.csv, accuracy: 0.04
Calculated for college_physics_test.csv, accuracy: 0.0
Calculated for virology_test.csv, accuracy: 0.030303030303030304
Calculated for high_school_psychology_test.csv, accuracy: 0.046296296296296294
Calculated for college_chemistry_test.csv, accuracy: 0.0
Calculated for us_foreign_policy_test.csv, accuracy: 0.05263157894736842
Calculated for prehistory_test.csv, accuracy: 0.015625
Calculated for high_school_geography_test.csv, accuracy: 0.07692307692307693
0.02885811785358698
```

Выводы:

1. Метод имеет место быть
Даже при наших небольших ресурсах удалось частично добиться хороших метрик

2. С увеличением количества слоев и датасета метрики улучшались

Простой и дешевый способ разработки **Small base LM** для последующего дообучения под определенную задачу

Inheritune

Благодарим за внимание