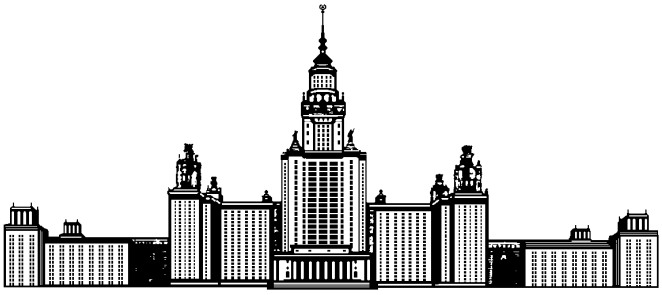

Московский государственный университет имени М. В. Ломоносова
Факультет вычислительной математики и кибернетики
Кафедра математических методов прогнозирования

Васильев Руслан Леонидович

Применение генеративных текстовых моделей для решения задач обработки естественного языка

Отчет по преддипломной практике

Научный руководитель:
д.ф-м.н., профессор
А. Г. Дьяконов

Москва, 2021

Содержание

1 Введение

// more info will be added tonight [**shavrina2020russiansuperglue**] [**li2021prefix**] [**liu2021gpt**] [**lester2021power**] [**gpt2**] [**gpt3**]

2 Постановка задачи

	Тип задачи	**Метрики качества**	**Train / Val / Test**
LiDiRus	Диагностика	Matthews Corr	0 / 0 / 1104
RCB	Логический вывод	Avg. F1 / Accuracy	438 / 220 / 438
PARus	Здравый смысл	Accuracy	400 / 100 / 500
MuSeRC	Машинное чтение	F1a / EM	500 / 100 / 322
TERRa	Логический вывод	Accuracy	2616 / 307 / 3198
RUSSE	Здравый смысл	Accuracy	19845 / 8508 / 18892
RWSD	Причинно-след. связь	Accuracy	606 / 204 / 154
DaNetQA	Знание	Accuracy	1749 / 821 / 805
RuCoS	Машинное чтение	F1 / EM	72193 / 7577 / 7257

Таблица 1: Задачи бенчмарка Russian SuperGLUE

Бенчмарк Russian SuperGLUE [**shavrina2020russiansuperglue**] состоит из 9 различных языковых задач.

// more info will be added tonight

- **LiDiRus**
- **RCB**
- **PARus**
- **MuSeRC**
- **TERRa**
- **RUSSE**
- **RWSD**
- **DaNetQA**
- **RuCoS**

Размеры датасетов и функционалы качества приведены в **??**, общий функционал качества — среднее значение метрик по всем задачам. Если для одного датасета приводится несколько метрик, то они сначала усредняются дла самой задачи, среднее считается по 9 итоговым значениям .

3 Метод решения

Метод решения основан на оптимизации несуществующих «токенов», играющих роль подводки, на относительно небольшом числе примеров.

4 Эксперименты

Существенная часть экспериментов с p-tuning при решении Russian SuperGLUE была связана с выбором оптимальной постановки задачи для нейросети: порядок исходных текстов (уточняющая информации, запросы, вопросы и ответы), выбор префикса и суффикса, наличие дополнительных подводок, разбиение на подзадачи.

Когда достаточно универсальный подход найден, а все особенности отдельных задач учтены, требовалось научиться находить оптимальные p-tune вектора. Впоследствии (поскольку методу не требуется много данных) процесс оптимизации стал достаточно быстрым — но прежде были проведены многочисленные эксперементы с параметризацией (размерности, наличие дополнительных слоев), непосредственно оптимизацией, регуляризацией и начальным приближением.

4.1 Параметризация

P-tuning допускает различные варианты параметризации блоков, основные типы:

- Только p-tune вектора (как эмбеддинги);
- Эмбеддинги $\rightarrow$ MLP $\rightarrow$ p-tune вектора;
- Эмбеддинги $\rightarrow$ LSTM $\rightarrow$ p-tune вектора;
- Эмбеддинги $\rightarrow$ LSTM $\rightarrow$ MLP $\rightarrow$ p-tune вектора;

Ключевыми гиперпараметрами здесь являются:

- Число p-tune векторов;
- Расположение блоков;
- Параметры MLP/LSTM (при наличии).

4.2 Оптимизация

Чтобы быстро сойтись к хорошему решению, важно учесть нюансы, связанные с применением градиентных алгоритмов оптимизации:

- Темп обучения и стратегия его изменения;
- Количество итераций/эпох;
- Размеры батча;
- Параметры оптимизатора.

4.3 Регуляризация

Многие задачи Russian SuperGLUE имеют небольшое число данных. Чтобы избежать переобучения под тренировачные данные, была использована отдельная валидационная подвыборка, позволяющая делать ранний останов. Тем не менее полезными оказались различные методы регуляризации, такие как:

- l_2-регуляризация на сами p-tune вектора;
- Дропауты на эмбеддинги, механизм внимания, полносвязные слои;

4.4 Инициализация

Чем меньше данных в задаче, тем большую роль играет инициализация p-tune блоков. Для задач наподобие RuCoS, где примеров десятки тысяч, эффект был почти не заметен, а вот для малочисленных (RCB, PARus, RWSD) от начального приближения могло сильно зависеть итоговое качество.

Основные исследованные стратегии:

- $\mathcal{N}(0, \sigma^2)$ с дисперсией инициализации эмбеддингов исходного трансформера;
- $\mathcal{N}(0, \sigma^2)$, где дисперсия совпадает с выборочной дисперсией обученных токенов;
- Осмысленной подводкой [**lester2021power**];
- Случайными токенами из словаря;
- Токенами, связанными с задачей (классами и т. д).

5 Результаты

Модель	Общий результат
Human Benchmark	0.811
Golden Transformer v2.0	0.755
YaLM	0.711
ruT5	0.686
ruRoberta	0.684

Таблица 2: Топ-5 решений задач бенчмарка Russian SuperGLUE

Модель	Архитектура	Общее число параметров
YaLM	decoder	3.3B млрд
ruT5	encoder-decoder	740 млн
ruRoberta	encoder	355 млн

Таблица 3: Общее число параметров, оптимизируемых во время предобучения

Модель	Метод дообучения	Число оптимизируемых параметров
YaLM	p-tuning	40 тыс
ruT5	finetuning	740 млн
ruRoberta	finetuning	335 млн

Таблица 4: Сравнение параметров при дообучении моделей на целевые задачи

6 Заключение

В рамках преддипломной практики были исследованы языковые модели — генеративные трансформеры, основанные на архитектуре GPT. Было произведено множество экспериментов с новым подходом дообучения текстовых трансформеров — P-tuning.

С помощью оптимизации небольшого числа параметров оказалось возможным решение самых разных задач, связанных с пониманием естественного языка. На наборе из 9 различных заданий на русском языке, собранных в бечнмарке Russian SuperGLUE, было получено лучшее качество среди single-model подходов.