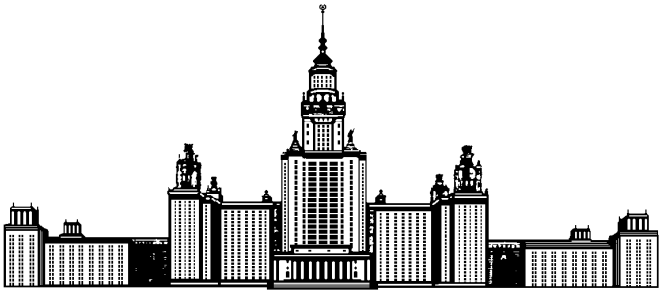

Московский государственный университет имени М. В. Ломоносова
Факультет вычислительной математики и кибернетики
Кафедра математических методов прогнозирования

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

Языковые модели, основанные на архитектуре трансформера [1, 3, 6, 4, 11], применяются в решении множества задач: перевод, генерация для диалогового ассистента, суммаризация текстов, получения ответа на вопрос и т.д. Для сравнения языковых моделей были собраны датасеты и бенчмарки (например, [5]), позволяющие оценить качество работы модели на различных задачах.

Обучение современных моделей разделено на два этапа: предобучение на неразмеченных текстах и дообучение на определенной задаче. Обычно для дообучения нейросетей веса нейросети оптимизируются на конкретной задаче (fine-tuning), также могут добавляться обучаемые «классификационные» слои на выходы или «адаптеры» [2], а случае многомиллиардных моделей можно обойтись и без дообучения [6, 4, 11].

Альтернативный подход — p-tuning — был предложен в этом году [9, 10, 8], его идея заключается в обучении дополнительных эмбеддингов контекста. В ходе преддипломной практики данный метод был исследован и применен для решении различных языковых задач, собранных в бенчмарке [7].

2 Постановка задачи

В данной работе в качестве основного набора задач был выбран Russian SuperGLUE — бенчмарк из 9 различных задач на русском языке.

	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 [7]

Кратко опишем каждую из задач.

- **LiDiRus** (Linguistic Diagnostic for Russian)

  Диагностический датасет, составленный идентично [5] и включающий в себя специфические примеры логического вывода;

- **RCB** (Russian Commitment Bank)

В паре предложений (текста и гипотезы из него) требуется определить, является ли гипотеза с точки зрения автора следствием / противоречием или не имеет ясной причинно-следственной связи с текстом.

- **PARus** (Plausible Alternatives for Russian)

  Задача на целеполагание: необходимо (в зависимости от примера) подобрать нужное следствие или причину к данному предложению.

- **MuSeRC** (Multi-Sentence Reading Comprehension) (Russian reading comprehension with Commonsense)

  Датасет состоит из текстов, к каждому из которых дан набор вопросов и потенциальных ответов на них — необходимо выбрать верные.

- **TERRa** (Textual Entailment Recognition for Russian)

  Стандартная задача логического вывода: выбор связи между двумя предложениями — следствие или противоречие.

- **RUSSE** (Russian Semantic Evaluation)

  Здесь проверяется понимание смысла: необходимо распознать, совпадает ли лексическое значение слова, использованного в паре предложений.

- **RWSD** (Russian Winograd Schema Dataset)

  Задания на логику: необходимо разрешить неопределенность, выбрав, к какой сущности относится местоимение.

- **DaNetQA**.

  Различные вопросы на здравый смысл с возможными ответами «да» и «нет».

- **RuCoS** (Russian reading Comprehension with Commonsense)

  Наконец, в данной задаче требуется по тексту и его резюме заполнить пропуск (выбрав из вариантов).

Размеры датасетов и функционалы качества приведены в табл. 1, 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

Впервые идея была предложена в [9], впоследствие получила развитие в [10, 11]. Метод решения основан на оптимизации несуществующих «токенов», играющих роль подводки, на относительно небольшом числе примеров.

Рассмотрим предобученную языковую модель $\mathcal{M}$ и последовательность токенов $\mathbf{x}_{1:n} = \{x_0, x_1, \ldots, x_n\}$, отображаемую в эмбеддинги $\{\mathbf{e}(x_0), \mathbf{e}(x_1), \ldots, \mathbf{e}(x_n)\}$ первым слоем модели $\mathcal{M}$.

Пусть $\mathbf{x}$ — известная посылка (например, вопрос к тексту или фрагмент рассказа), а $\mathbf{y}$ — искомый текст (например, ответ на вопрос или краткое содержание фрагмента). С языковой моделью такие задачу можно решаются достаточно общим подходом:

$$\{\mathbf{e}([\mathrm{P}_{0:i}]), \mathbf{e}(\mathbf{x}), \mathbf{e}([\mathrm{P}_{i+1:m}]), \underbrace{\mathbf{e}(\mathbf{y})}_{?}\}, \tag{1}$$

где $\mathrm{P}_{0:i}$ и $\mathrm{P}_{i+1:m}$ — формулировка задачи (например, «Прочитай текст и ответь на вопрос», «Короче говоря:»).

Вместо подбора удачной формулировки p-tuning настраивает ее с помощью псевдотокенов:

$$\{h_0, \ldots, h_i, \mathbf{e}(\mathbf{x}), h_{i+1}, \ldots, h_m, \mathbf{e}(\mathbf{y})\}, \tag{2}$$

где $h_0, \ldots, h_m$ — обучаемые эмбеддинги, остальные параметры сети остаются прежними.

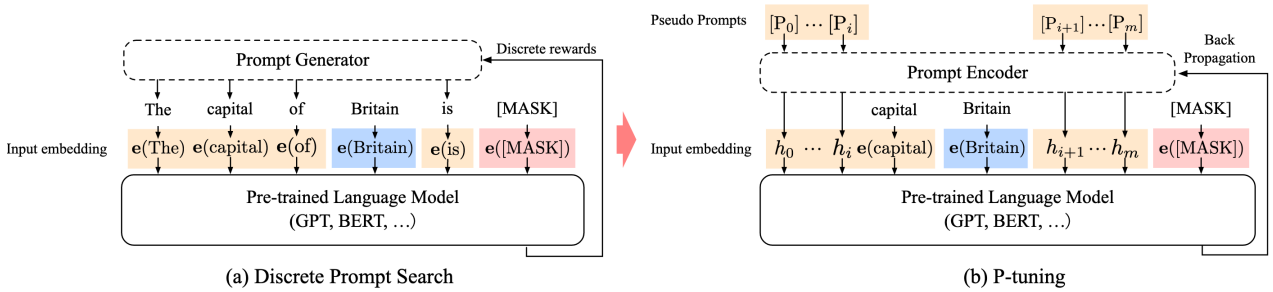

Рис. 1: Вместо выбора оптимальных слов для описания задачи [10] (дискретное множество) производится оптимизация «подводки» в непрерывном пространстве: модель обучается предсказывать суффикс на основе контекста, часть которого — настраиваемые p-tune вектора.

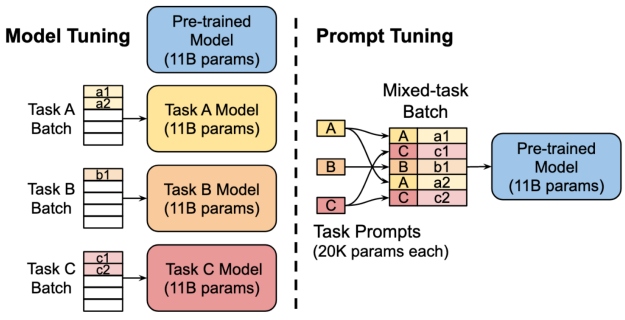

Рис. 2: Преимущество работы p-tuning на инференсе [8]: вместо отдельных обученных на каждую задачу моделей разные p-tune вектора могут подставляться динамически к самим текстам (их представлению), модель при этом остается одна и та же.

В качестве предобученных трансформеров были использованы модели семейства YaLM, близкие по архитектуре к [3, 6].

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 [8];
- Случайными токенами из словаря;
- Токенами, связанными с задачей (классами и т. д).

5 Результаты

В YaLM p-tuning превзошел предыдущие варианты дообучения (fine-tuning, fewshot) на исследуемых моделях. На тестовых данных Russian SuperGLUE решение в течение некоторого времени лидировало среди всех ML-подходов. Сейчас (декабрь 2021, табл. 2) оно уступает только ансамблю из трансформеров, в котором основной выигрыш был получен за счет использования обученных моделей на английском языке и машинный перевод заданий бенчмарка.

Тем не менее p-tuning (табл. 4) пока остается SOTA на лидерборде среди single-model решений. Отметим, что важную роль в этом сыграло и качество самой предобученной языковой модели табл. 3. Качество на отдельных задачах приведено в табл. 5 и табл. 6.

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, декабрь 2021

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	fine-tuning	740 млн
ruRoberta	fine-tuning	335 млн

Таблица 4: Сравнение параметров при дообучении моделей на целевые задачи

Модель	LiDiRus	RCB	PARus	MuSeRC
YaLM	**0.364**	0.357 / 0.479	**0.834**	**0.892 / 0.707**
ruT5	0.320	**0.450 / 0.532**	0.764	0.855 / 0.608
ruRoberta	0.343	0.357 / 0.518	0.722	0.861 / 0.630

Таблица 5: Сводка по отдельным задачам RSG, декабрь 2021, часть 1

Модель	TERRa	RUSSE	RWSD	DaNetQA	RuCoS
YaLM	**0.841**	0.710	**0.669**	**0.85**	**0.92 / 0.916**
ruT5	0.775	**0.773**	**0.669**	0.79	0.86 / 0.859
ruRoberta	0.801	0.748	**0.669**	0.82	0.87 / 0.867

Таблица 6: Сводка по отдельным задачам RSG, декабрь 2021, часть 2

6 Заключение

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

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

Список литературы

1. *Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A. N., Kaiser Ł., Polosukhin I.* Attention is all you need // Advances in neural information processing systems. — 2017. — P. 5998–6008.
2. *Houlsby N., Giurgiu A., Jastrzebski S., Morrone B., De Laroussilhe Q., Gesmundo A., Attariyan M., Gelly S.* Parameter-efficient transfer learning for NLP // International Conference on Machine Learning. — PMLR. 2019. — P. 2790–2799.
3. *Radford A., Wu J., Child R., Luan D., Amodei D., Sutskever I.,* [et al.]. Language models are unsupervised multitask learners // OpenAI blog. — 2019. — Vol. 1, no. 8. — P. 9.
4. *Raffel C., Shazeer N., Roberts A., Lee K., Narang S., Matena M., Zhou Y., Li W., Liu P. J.* Exploring the limits of transfer learning with a unified text-to-text transformer // arXiv preprint arXiv:1910.10683. — 2019.
5. *Wang A., Pruksachatkun Y., Nangia N., Singh A., Michael J., Hill F., Levy O., Bowman S. R.* Superglue: A stickier benchmark for general-purpose language understanding systems // arXiv preprint arXiv:1905.00537. — 2019.
6. *Brown T. B., Mann B., Ryder N., Subbiah M., Kaplan J., Dhariwal P., Neelakantan A., Shyam P., Sastry G., Askell A.,* [et al.]. Language models are few-shot learners // arXiv preprint arXiv:2005.14165. — 2020.
7. *Shavrina T., Fenogenova A., Emelyanov A., Shevelev D., Artemova E., Malykh V., Mikhailov V., Tikhonova M., Chertok A., Evlampiev A.* RussianSuperGLUE: A Russian Language Understanding Evaluation Benchmark // arXiv preprint arXiv:2010.15925. — 2020.
8. *Lester B., Al-Rfou R., Constant N.* The power of scale for parameter-efficient prompt tuning // arXiv preprint arXiv:2104.08691. — 2021.
9. *Li X. L., Liang P.* Prefix-tuning: Optimizing continuous prompts for generation // arXiv preprint arXiv:2101.00190. — 2021.
10. *Liu X., Zheng Y., Du Z., Ding M., Qian Y., Yang Z., Tang J.* GPT Understands, Too // arXiv preprint arXiv:2103.10385. — 2021.
11. *Rae J. W.* [et al.]. Scaling Language Models: Methods, Analysis & Insights from Training Gopher // arXiv preprint arXiv:2112.11446. — 2021.