ДОКЛАДЫ РОССИЙСКОЙ АКАДЕМИИ НАУК. МАТЕМАТИКА, ИНФОРМАТИКА, ПРОЦЕССЫ УПРАВЛЕНИЯ, 2023, том 514, № 2, с. 375–384

УДК 004.8

БЕЗОПАСНОЕ ПРЕДОБУЧЕНИЕ ГЛУБОКИХ ЯЗЫКОВЫХ МОДЕЛЕЙ НА СИНТЕТИЧЕСКОМ ПСЕВДОЯЗЫКЕ

© 2023 г. Т. Е. Горбачева[1,*], И. Бондаренко[1,]**

Представлено академиком РАН А.Л. Семеновым
Поступило 03.09.2023 г.
После доработки 15.09.2023 г.
Принято к публикации 24.10.2023 г.

В данной работе проводится сравнение предварительного обучения трансформера на текстах естественного языка и на предложениях синтетического псевдоязыка. Искусственные тексты были автоматически сгенерированы по написанным нами правилам в контекстно-свободной грамматике. Результаты дообучения на выполнение заданий проекта RussianSuperGLUE статистически достоверно показали, что модели имеют одинаковые оценки, т.е. можно считать, что использование искусственных данных дает преимущество для “безопасности” искусственного интеллекта за счет возможности полностью контролировать состав выборки. Также мы можем говорить о том, что на этапе предобучения модели типа RoBERTa достаточно научиться распознавать только синтаксические и морфологические закономерности языка, которые могут быть успешно созданы довольно таким простым способом, как контекстно-свободная грамматика.

Ключевые слова: методы глубокого обучения, трансформеры, предварительное обучение, автоматическое создание текста, глубокие языковые модели, синтетические данные, “безопасность” нейросети

DOI: 10.31857/S2686954323601860, **EDN:** CYAWMI

1. ВВЕДЕНИЕ

Глубокие нейронные сети за последнее время стали наиболее популярным инструментом для решения большинства задач искусственного интеллекта и особенно задач анализа и генерации текстов на естественном языке, относящихся к т.н. “разговорному искусственному интеллекту”. Это произошло по двум причинам:

• нейронная сеть строит обучаемую иерархию представлений;

• эта иерархия представлений является переиспользуемой между задачами, на чем основана известная техника переноса обучения, когда нейросетевая модель предварительно обучается (предобучается) на большой обучающей выборке решать ненужную задачу, для которой доступна “дешевая” или автоматическая разметка, а потом дообучается на малой обучающей выборке, описывающей конечную задачу и размеченной вручную.

При этом глубокие нейронные сети, как и другие методы машинного обучения, могут быть неустойчивы к ряду уязвимостей и угроз, что создает препятствия при создании доверительного искусственного интеллекта на базе нейросетевого подхода.

Существует ряд ключевых направлений работ по обеспечению доверия к нейросетевым системам искусственного интеллекта, включающий в себя как традиционные способы обеспечения безопасности информационных систем (например, создание доверенных фреймворков машинного обучения), так и особые подходы применительно к системам искусственного интеллекта как отдельному виду информационных систем со своими специфическими уязвимостями и угрозами (более полно такие направления работ перечислены, например, в [1]).

Среди специфических угроз нейросетевым системам искусственного интеллекта следует отметить неустойчивость к изменению распределения обучающей выборки, что может обеспечить внедрение закладок в обученную на такой выборке нейросеть [2]. И если доступ к малой обучающей выборке для дообучения можно контролировать, а “отравление” такой выборки – выявить, то с гигантскими обучающими выборками для предобучения это сделать практически невозможно.

[1]*Новосибирский государственный университет, Новосибирск, Россия*

**E-mail: t.gorbacheva@alumni.nsu.ru*

***E-mail: i.bondarenko@g.nsu.ru*

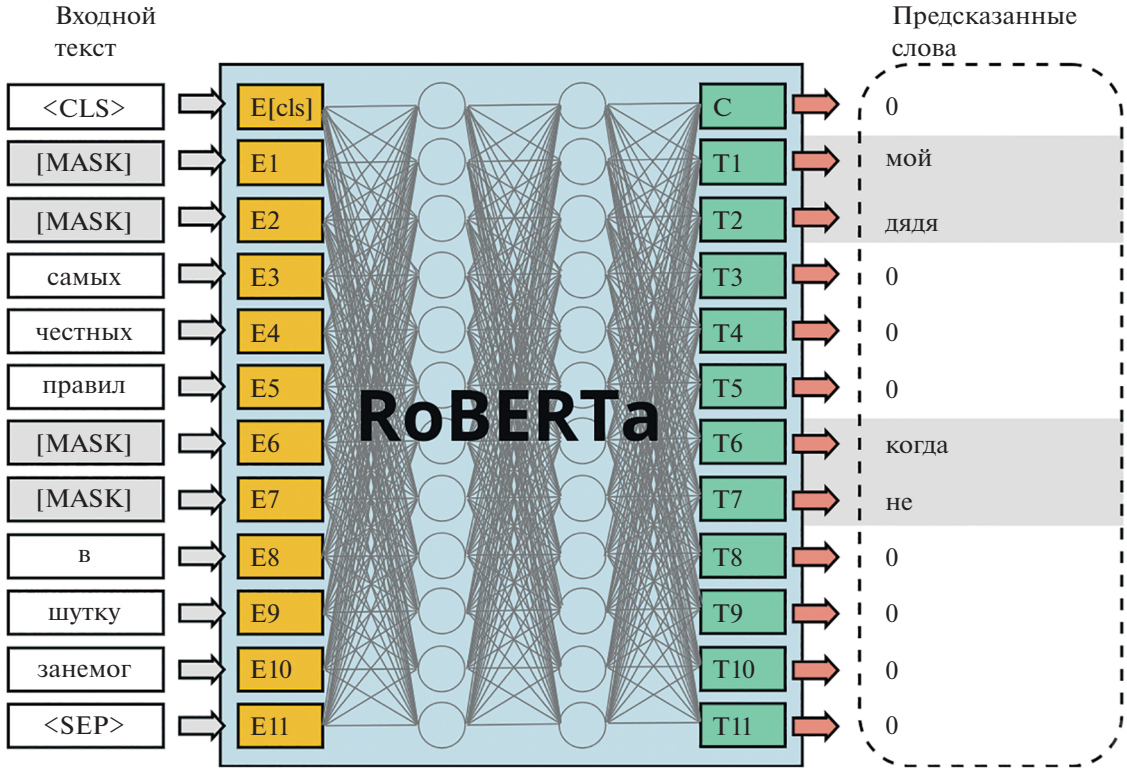

Рис. 1. Пример решения задачи восстановления замаскированных токенов трансформерной нейросетью-кодировщиком типа RoBERTa (здесь токены <CLS> и <SEP> являются специальными лексическими единицами, обозначающими начало и конец текста, а токен [MASK] используется для маскирования во входном тексте тех слов, которые должна предсказать нейросеть).

Единственный способ гарантировать отсутствие “закладок” в обучающей выборке для предобучения – это автоматический синтез такой выборки на основе специальной модели порождения данных и произвольного начального значения генератора случайных чисел, известного только создателю выборки. В задачах компьютерного зрения подход с использованием синтетических данных для предобучения (вместо популярного ранее датасета ImageNet, известного в том числе и своими проблемами) уже применяется – например, синтетические изображения могут быть сгенерированы с помощью фракталов [3]. Но в задачах разговорного искусственного интеллекта подобный подход не используется. Цель данной работы – восполнить этот пробел хотя бы для трансформерных нейросетей-кодировщиков типа RoBERTa [4] решив следующие задачи:

1) задачу создания простого и прозрачного метода синтеза большого текстового корпуса для предобучения, целью которого является восстановление замаскированных токенов (Masked Language Modeling, или MLM);

2) задачу анализа эффективности предобучения трансформерной нейросети-кодировщика на синтетическом текстовом корпусе по сравнению с предварительным обучением на “натуральном” текстовом корпусе аналогичного размера.

2. СОЗДАНИЕ СИНТЕТИЧЕСКОГО ТЕКСТОВОГО КОРПУСА

Известно, что основной задачей, решению которой предобучается трансформерная нейросеть-кодировщик перед ее дообучением конкретным задачам анализа текстов, является задача восстановления замаскированных токенов (Masked Language Modeling, или MLM) в тексте на естественном языке. Так, для предобучения нейрости типа BERT эта задача является одной из двух, а для предобучения нейросети типа RoBERTa – вообще единственной задачей, как, например, показано на рис. 1.

Возможность решения такой задачи нейронной сетью основана на том, что естественный язык обладает синтаксической структурой, а значит, связи между словами даже в словосочетаниях и простых предложениях подчиняются некоторым грамматическим правилам. На основе этого был предложен метод синтеза текстового корпуса для предобучения, основанный на следующем:

1) создается формальная грамматика подмножества естественного языка, в которой алфавит терминальных символов формируется на основе лингвистических тезаурусов типа WordNet;

2) на основе грамматики генерируются предложения как синтаксические единицы языка,

причем выбор конкретного терминального символа из алфавита выполняется генератором псевдослучайных чисел;

3) в результате формируется текстовый корпус, состав которого определяется только формальной грамматикой, заданным количеством предложений и начальным значением генератора псевдослучайных чисел.

Для генерации текстов была использована контекстно-свободная грамматика. Такой подход обладает рядом преимуществ.

Во-первых, в отличие от различных генеративных трансформеров, контекстно-свободная грамматика является полностью предсказуемой и, соответственно, более "безопасной" системой.

Во-вторых, такой способ менее сложен вычислительно по сравнению, например, с цепями Маркова.

В-третьих, было выдвинуто предположение, что в текстовом корпусе для предобучения нейросетевых моделей языка достаточным является моделирование наиболее простых синтаксических правил, в то время как получение "знаний" о семантике языка нейросетью происходит на этапе дообучения конечной задаче, носящей более семантический характер. Это предположение было основано на том, что глубокая нейронная сеть – это иерархия обучаемых представлений, выстраиваемая в направлении "от простого к сложному": первые слои нейронной сети моделируют весьма простые представления, общие для широкого класса задач, а чем ближе к последнему слою, тем более сложными, специфичными для конкретной решаемой задачи, становятся представления. И при переносе обучения на новую задачу наиболее полезными оказываются не последние, специфичные для конкретной задачи, представления, а те более общие представления, которые формируются начальными и средними слоями. В лингвистике же более простыми уровнями представления языка являются морфология и синтаксис, более сложным (и более специфичным) уровнем считается семантика, далее – прагматика, и т.п. Таким образом, на этапе предобучения нейросетевой модели языка наиболее полезными представлениями, приобретаемыми обучаемой моделью, являются синтаксические, поэтому именно синтаксис должен быть представлен в синтезируемом текстовом корпусе.

Для описания правил контекстно-свободной грамматики использовался формат JSGF (Java Speech Grammar Format или JSpeech Grammar Format) [5].

В качестве языка, для которого будет происходить моделирование, был выбран русский язык. В разработке контекстно-свободного "псевдорусского" языка мы руководствовались морфологическими и синтаксическими правилами русского языка, составление происходило с опорой на материал справочника русского языка Баранова М.Т. и др., составленного под редакцией Н.М. Шанского [6], без учета семантики.

Кроме того, при составлении правил не учитывались пунктуация и употребление больших букв в начале предложений и для имен собственных.

Для удобства созданные правила были сформированы в несколько групп:

- так называемая "группа подлежащего", состоит из правил подлежащего, согласованного и несогласованного определений;
- "группа сказуемого", состоит из правил простого или составного сказуемого и обстоятельства образа действия;
- "группа дополнения", состоит из правил дополнения, согласованного и несогласованного определений;
- "группа обстоятельства", состоит из правил деепричастия и обстоятельства места и времени.

Помимо этих групп, были созданы правила для частиц, предлогов и союзов. Все элементы правил согласуются между собой. Каждое правило для члена предложения может быть представлено одним или несколькими правилами определенных частей речи. Более детальное описание грамматики с рядом примеров доступно для ознакомления в Приложении 1. Созданные правила размещены на GitHub [1].

Для заполнения словаря терминалов контекстно-свободной грамматики были необходимы слова русского языка с указанием их морфологической информации. Небольшая часть необходимых данных была взята из справочника русского языка Баранова М.Т. и др. [6], а для получения информации о большинстве слов использовались два набора данных: тезаурус RuWordNet [7] и корпус "Тайга" [8]. RuWordNet представляет собой вручную размеченные синсеты, однако только для трех частей речи (существительных, прилагательных и глаголов), причем каждая представлена лишь начальной формой. Для получения всех грамматических форм с указанием морфологической информации слова мы использовали морфологический анализатор pymorphy2 [10]. "Тайга" является большой подборкой данных, но они были размечены автоматически парсером UDPipe [11]. Для валидации авторазметки "Тайги" применялся следующий подход: морфологическая информация в "Тайге", сгенерированная UDPipe, сравнивалась с той, которую определяет парсер Spacy [12] с русско-

[1] https://github.com/GorbachevaTaisia/JSGF_generative_grammar

языкой моделью [2], и затем в алфавит терминалов добавлялись только те слова, в которых авторазметка двух систем совпадает.

Из “Тайги” было выбрано шесть под-корпусов (Lenta.ru, Интерфакс, N+1, Facebook, Vkontakte, Twitter). Из каждого корпуса брались слова, написанные только кириллицей. Под-корпуса социальных сетей также потребовали дополнительной чистки от опечаток и ругательств. Таким образом, после чистки корпуса социальных сетей, объединения всех корпусов “Тайги” с грамматическими парадигмами слов из тезауруса RuWordNet общий объем алфавита терминалов разработанной контекстно-свободной грамматики составил 2 477 009 уникальных слов.

Для генерации по правилам использовалась система JSGFTools [13]. Данный система обладала некоторыми недостатками, которые потребовали доработок. Часть из них была внесена в саму систему, а другая часть – в правила созданной нами грамматики. Исправленная нами версия JSGFTools доступна на GitHub[3].

Было принято решение, что датасет текстов нашего синтетического псевдоязыка должен содержать два миллиона предложений. Такое число показалось нам оптимальным в вопросе полноты предобучения модели и затраченного на это времени. Ниже приведем несколько предложений полученной выборки (сохранено оригинальное отсутствие пунктуации и больших букв):

1) либерализуем мы хоронив между согреваньем;

2) не предо календариком обкидавши и я измыслить укрощать поглубже б желал;

3) и вы исключительно устыдив собираетесь фырчать милитаризировавши;

4) почти затупив не виноватого сто первого преуспевания почти гриппуя шестиклассник с краснознаменском дидактиков сто десятые ваши даже и рекламировать выболтать принимают все-таки навернув;

5) будто отпиливаемой типично становишься ли ты гофрирования;

6) вот я чванюсь;

7) едва ли заскрипевши справа от коллекции уложить решало убийственно осчастливливание видеобращения спешившее почти с мини-футбол просачиваясь;

8) по врачам подговаривавши годимся уж мы какие-то;

9) и на туринск воюя и бездетность при приволжье распоясавшись выясняла;

10) словно вы называетесь не вы.

Из приведенных примеров можно заметить, что полученные конструкции отличаются разнообразной синтаксической и морфологической структурой. Помимо этого, предложения корректны с точки зрения морфологических и синтаксических правил русского языка. Следует пояснить, почему нами не учитывается пунктуация и расстановка прописных букв. Целью постановки знаков препинания и больших букв является организация письменного текста для облегчения его понимания. Такую задачу можно больше отнести к семантике, однако на данном этапе исследования мы ее не учитывали.

3. ПРЕДОБУЧЕНИЕ ГЛУБОКОЙ ЯЗЫКОВОЙ МОДЕЛИ

Для проведения в дальнейшем сравнения было предобучено две модели типа RoBERTa: одна на полученной выборке искусственных предложений и другая на выборке “естественных” текстов. Полученные языковые модели размещены на Hugging Face[4].

Для получения датасета из двух миллионов “естественных” предложений мы использовали два источника данных: новостные под-корпуса “Тайги” (Lenta.ru, Интерфакс, N+1) [8] и тексты русской Википедии. При выборе предложений учитывались особенности сгенерированных текстов, а именно не использовались предложения, содержащие:

- числительные, написанные цифрами;
- только подлежащее без сказуемого, выраженного глаголом;
- какие-либо символы помимо кириллических.

Предварительное обучение обеих моделей происходило аналогично. Подробное описание параметров запусков находится в Приложении 2.

Для удобства введем следующие обозначения: NaturalRoBERTa – трансформер RoBERTa, предобученный на предложениях естественного языка; SyntheticRoBERTa – трансформер RoBERTa, предобученный на предложениях синтетического псевдоязыка. Предобучение NaturalRoBERTa было завершено на 34 эпохе со значением функции потерь, равным 8.7587. SyntheticRoBERTa завершила свое предобучение на 12 эпохе со значением функции потерь 8.6713. На рис. 2 представлены графики зависимости значения функции потерь на обучающей и “валидационной” выборках от эпохи обучения моделей: можно видеть, что нейросети демонстрируют практически одинаковый результат во время предварительного обучения.

[2] https://spacy.io/models/ru
[3] https://github.com/GorbachevaTaisia/JSGFTools__
[4] https://huggingface.co/tay-yozhik/NaturalRoBERTahttps://huggingface.co/tay-yozhik/SyntheticRoBERTa

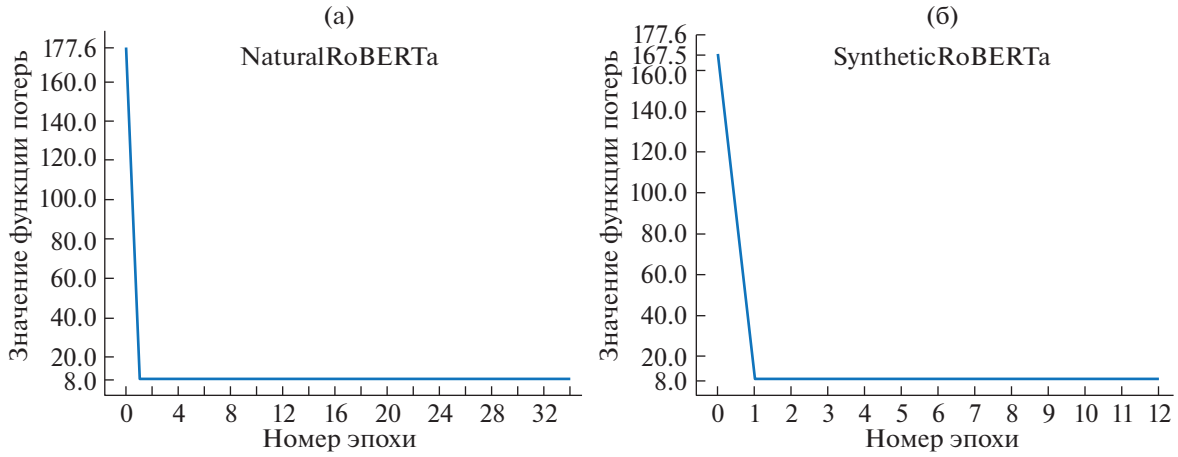

Рис. 2. Значения функции потерь во время предобучения трансформеров на предложениях естественного языка (а) и на предложениях синтетического псевдоязыка (б).

4. ЭКСПЕРИМЕНТЫ С ДООБУЧЕНИЕМ

Для тестирования и сравнения полученных моделей был использован задачник Russian SuperGLUE [9]. Тесты из этого задачника позволяют оценить качество любой нейросетевой модели, предобученной на русскоязычном текстовом корпусе, путем дообучения этой модели выполнению девяти задач, таких как понимание логической связи между двумя фразами, разрешение полисемии слов с учетом их контекста, определение истинности или ложности высказываний, и т.п. В целом данный задачник охватывает широкий круг задач на понимание естественного языка (Natural language understanding, NLU).

В табл. 1 приведены общие результаты тестирования моделей NaturalRoBERTa и SyntheticRoBERTa. Также приведены результаты известной и ранее апробированной модели RuGPT3Small – нейросетевой модели языка, которая обладает наиболее близким количеством параметров среди всех моделей, представленных в Russian SuperGLUE – в решении аналогичных задач. Следует заметить, что модель NaturalRoBERTa демонстрирует результаты, приблизительно равные результатам RuGPT3Small, что исключает возможные ошибки предобучения NaturalRoBERTa и позволяет использовать ее как эталон при сравнении с SyntheticRoBERTa.

Для того, чтобы статистически достоверно сравнить все полученные результаты тестирования моделей, был использован знаковый критерий Вилкоксона (Уилкоксона). В качестве одной выборки рассматривались экспериментальные результаты модели NaturalRoBERTa, а в качестве другой выборки – результаты модели SyntheticRoBERTa. Проверялись следующие гипотезы:

- Нулевая гипотеза H_0: математические ожидания двух выборок равны;
- Альтернативная гипотеза H_1: математические ожидания двух выборок различны.

Было получено значение статистики теста 1.0, соответствующее двустороннее значение p равно 0.14413, т.е. нулевая гипотеза подтвердилась, и принципиальных различий между NaturalRoBERTa и SyntheticRoBERTa в способности анализировать тексты на естественном (русском) языке обнаружено не было.

Таблица 1. Значения метрик, полученные при тестировании NaturalRoBERTa, SyntheticRoBERTa и RuGPT3Small в некоторых заданиях RussianSuperGLUE

Задание	NaturalRoBERTa	SyntheticRoBERTa	RuGPT3Small	Метрика
LiDiRus	0.0	0.0	–0.013	Коэффициент Мэтью
RCB	0.217/0.484	0.091/0.158	0.356/0.473	F1/Точность
PARus	0.498	0.502	0.562	Точность
TERRa	0.487	0.487	0.488	Точность
RUSSE	0.587	0.587	0.57	Точность
RWSD	0.669	0.331	0.669	Точность

5. ВЫВОДЫ

Предложенный в данной работе метод синтеза текстового корпуса на основе контекстно-свободной грамматики показал свою эффективность: трансформерная нейросеть-кодировщик типа RoBERTa предобучалась на синтетических текстах не менее эффективно, чем на “натуральных” (критерием эффективности здесь выступала способность предобученной модели дообучаться на открытом наборе разнообразных задач анализа естественного языка Russian SuperGLUE). Этот результат демонстрирует возможность построения более безопасного (доверенного) разговорного искусственного интеллекта, чем это было ранее, причем безопасность использования синтетических текстов для предобучения достигается одновременно как за счет формирования простого и проверяемого набора правил грамматики, так и за счет применения генератора псевдослучайных чисел для выбора терминальных символов. Но помимо этого, данный результат дает определенное основание для следующих рассуждений.

Во-первых, хотя естественный язык не является контекстно-свободным языком, его синтаксис может быть приближен контекстно-свободной грамматикой в той степени, которая достаточна для предобучения глубокой нейронной сети.

Во-вторых, на этапе предобучения можно считать несущественным то, что семантика естественного языка (не говоря уже о прагматике) практически не может быть отражена моделью на основе контекстно-свободной грамматики со случайным выбором порождаемого варианта. Видимо, те знания, которые приобретает нейросетевая модель языка, обучаясь задаче восстановления замаскированных слов (MLM), носят принципиально синтаксический характер, а семантические представления формируются у такой нейросети преимущественно на этапе дообучения конечной задаче (разумеется, если эта задача является задачей семантического анализа текста, как это имеет место в Russian SuperGLUE).

Тем не менее открытым остается следующий интересный вопрос: принципиально синтаксичными являются все типы предобучения нейросетевых моделей языка, или же данный вывод можно сделать только для ограниченного класса нейросетей-кодировщиков, которые предобучаются исключительно задаче MLM? Существуют ведь еще и модели типа “кодировщик-декодировщик”, такие как BART и T5, а также декодировщики, наиболее яркими представителями которых является семейство GPT. Модели типа “кодировщик-декодировщик” предобучаются, как правило, задаче подавления различных шумов в текстах, а декодировщики – продолжению текстов. Если знания, приобретаемые такими моделями на стадии предобучения, тоже в большей степени синтаксичны, то предложенный в данной статье метод использования контекстно-свободной грамматики для синтеза обучающих текстов обеспечивает безопасность предобучения и для этих моделей. В противном случае можно полагать, что такой метод не подходит для предубучения моделей этих типов вообще. Но ответ на данный вопрос находится вне рамок данной работы и требует дополнительного исследования

Приложение

1. ОПИСАНИЕ СОЗДАННЫХ ПРАВИЛ СИНТЕТИЧЕСКОГО ПСЕВДОЯЗЫКА

Приведем примеры правил для каждой группы и дадим комментарий.

“Группа подлежащего”. В грамматике представлены правила для каждого рода и числа подлежащего. Также учитывается лицо, это необходимо для удобства работы со временами сказуемого в дальнейшем.

На рис. 3 приведены некоторые правила для одного из типов подлежащего. Здесь

<agreed_attribute_s_masculine> представляет собой правило для согласованного определения к подлежащему, *<masculine_nouns>* – для существительного в мужском роде единственном числе именительном падеже, выполняющее роль подлежащего, *<inconsistent_attribute_s>* – для несогласованного определение к подлежащему. Согласно синтаксису формата грамматики JSGF, элементы в квадратных скобках являются не обязательными.

Поскольку в модуле JSGFTools на настоящий момент не реализована поддержка оператора замыкания Клини, считаем, что использование согласованного определения ноль, один или два раза является компромиссным решением, позволяющим отразить в грамматике особенности русского предложения без усложнения работы самого модуля.

Каждое правило, которое используется для задания части речи, представляет собой список слов вида *<masculine_nouns> = (слон|хомяк|кот);*, т.е. так называемый набор альтернатив.

Для получения других порядков слов в группе подлежащего производятся все возможные перестановки элементов *[<agreed_attribute_s_masculine>]*, *<masculine_nouns>* и *[<inconsistent_attribute_s>]*.

Правила для других типов подлежащих задаются аналогично. Подлежащее во множественном числе может выражаться не только существительным во множественном числе, но и сочетанием вида *(<masculine_nouns>|<feminine_nouns>|<neuter_nouns>) с <ablative_nouns>*.

```
<subject_group_direct_order_masculine> =
[<agreed_attribute_s_masculine>] <masculine_nouns>
[<inconsistent_attribute_s>];


<agreed_attribute_s_masculine_one> = (<masculine_adjectives>|
<masculine_pronouns>| <masculine_participles>| <masculine_numerals>);


<agreed_attribute_s_masculine> = (<agreed_attribute_s_masculine_one>|
<agreed_attribute_s_masculine_one>
<agreed_attribute_s_masculine_one>);
```

Рис. 3. Правила для подлежащего в мужском роде, единственном числе, 3 лице, с прямым порядком слов.

```
<predicate_group_transitive_second_person> = (

[<adverbial_pr>] <transitive_verbs_second_person>|

<transitive_verbs_second_person> [<adverbial_pr>]|

<auxiliary_transitive_verbs_second_person> <infinitives>
[<adverbial_pr>]|

<auxiliary_transitive_verbs_second_person> [<adverbial_pr>]
<infinitives>|

<infinitives> <auxiliary_transitive_verbs_second_person>
[<adverbial_pr>]|

<infinitives>[<adverbial_pr>]
<auxiliary_transitive_verbs_second_person>|

[<adverbial_pr>] <auxiliary_transitive_verbs_second_person>
<infinitives>|

[<adverbial_pr>] <infinitives>
<auxiliary_transitive_verbs_second_person>|[

<adverbial_pr>] <imperative_transitive_verbs_second_person>|

<imperative_transitive_verbs_second_person> [<adverbial_pr>]);
```

Рис. 4. Правило для сказуемого, выраженного переходным глаголом во 2 лице, единственном числе, настоящем или будущем времени, изъявительном или повелительном наклонении.

“Группа сказуемого”. В грамматике представлены правила согласно сказуемому, которое может быть простым, составным глагольным или составным именным.

На рис. 4 пример правила для сказуемого одного из видов. Здесь <*adverbial_pr*> представляет собой правило для обстоятельства образа действия, <*transitive_verbs_second_person*> – для переходного глагола во 2 лице <*auxiliary_transitive_verbs_second_person*> – для переходного вспомогательного глагола во 2 лице, единственном числе, настоящем или будущем времени, изъявительном наклонении, <*infinitives*> – для инфинитива, <*imperative_transitive_verbs_second_person*> – для переходного глагола во 2 лице, единственном числе, повелительном наклонении.

Как и с согласованными определениями, ввиду особенностей модуля для генерации, пришлось искать альтернативу оператору замыкания Клини.

```
<object_group_direct_order_genetive> =
([<agreed_attribute_o_masculine_genetive>] <masculine_nouns_genetive>|
[<agreed_attribute_o_feminine_genetive>] <feminine_nouns_genetive>|
[<agreed_attribute_o_neuter_genetive>] <neuter_nouns_genetive>|
[<agreed_attribute_o_plural_genetive>] <plural_nouns_genetive>)
[<inconsistent_attribute_o>];
```

Рис. 5. Правило для дополнения в родительном падеже с прямым порядком слов (другие падежи задаются аналогично).

Можно заметить, что в правиле отражены сразу все возможные порядки. Каждое правило, представляющее часть речи, как и в группе подлежащего, задается с использованием списка слов и вертикальной черты. Правила для других типов подлежащих задаются аналогично. Для получения условного наклонения к правилам с глаголами в прошедшем времени добавляется *[<particles_for_conditional>]*, задающее частицы *бы* и *б*. Составное именное сказуемое получается сочетанием правил вспомогательного глагола и именной части, которая получена при помощи существительного в творительном падеже или союзов как, будто, словно, точно с существительным, прилагательным, местоимением, причастием или числительным в именительном падеже.

“Группа дополнения”. В грамматике представлены правила для дополнения в одном из пяти косвенных падежей. Рассмотрим пример на рис. 5. Здесь *<agreed_attribute_o_masculine_genetive>*, *<agreed_attribute_o_feminine_genetive>*, *<agreed_attribute_o_neuter_genetive>*,

<agreed_attribute_o_plural_genetive> являются правилами для согласованного обстоятельства;

<masculine_nouns_genetive>, *<feminine_nouns_genetive>*, *<neuter_nouns_genetive>*,

<plural_nouns_genetive> представляют собой правило дополнения, а *<inconsistent_attribute_o>* соответствует несогласованному обстоятельству. Все эти правила задаются аналогично схожим правилам в группе существительных. Другие порядки получаются перестановкой элементов в

<object_group_direct_order_genetive>.

“Группа обстоятельства”. На рис. 6 приведен пример правила для обстоятельства с одним из порядков группы. Здесь *<gerunds>*является правилом деепричастия, *<particles>* – правилом для частицы, а *<adverbial_modifiers_of_time_place>* – правилом для обстоятельства места и времени, которое задается через предлог с семантикой места или времени и существительного в соответствующем падеже. Другие порядки получаются опять же перестановкой элементов.

Правила, из которых формируются предложения. Их можно разделить на две группы: предложения с переходным глаголом, который требует употребления дополнения в винительном падеже, и предложения с непереходным глаголом, для которого необходимо сочетание предлога и дополнения в соответствующем падеже. Количество правил соотносится с морфологическими характеристиками сказуемого и числом перестановок, необходимых для генерирования всех возможных порядков слов.

Рассмотрим пример на рис. 7. Можно видеть, что в группе подлежащего происходит выбор одного из порядков, аналогично в группах сказуемого, дополнения и обстоятельства. Обстоятельство и частица не имеют фиксированного положения внутри предложения и могут находиться в любом месте между другими группами.

Для генерирования дополнения переходного глагола используются сочетания правил предлога и группы дополнения в определенном падеже, например:

<prepositions_genetive> (<object_group_direct_order_genetive> |

<object_group_indirect_order_1_genetive>| object_group_indirect_order_2_genetive>).

Для начала генерации с помощью приписки *public* создано активное правило, содержащее набор альтернатив из всех правил для формирования предложения.

2. ПАРАМЕТРЫ ЗАПУСКА ПРЕДВАРИТЕЛЬНОГО ОБУЧЕНИЯ

Для предобучения обеих моделей задавались одинаковые алгоритмы и параметры.

В качестве токенизатора использовался токенизатор байтового уровня Byte-level byte pair encoding (BBPE) из библиотеки Transformers[5]. При его запуске были указаны следующие параметры:

[5] https://huggingface.co/docs/transformers/index

```
<adverbial_group_1> = [<particles>]
<adverbial_modifiers_of_time_place> <gerunds>;


<adverbial_modifiers_of_time_place> =
(<prepositions_genetive_of_time_place>
(<masculine_nouns_genetive>| <feminine_nouns_genetive>|
<neuter_nouns_genetive>| <plural_nouns_genetive>)|

<prepositions_dative_of_time_place> (<masculine_nouns_dative>|
<feminine_nouns_dative>| <neuter_nouns_dative>|
<plural_nouns_dative>)|

<prepositions_accusative_of_time_place>
(<masculine_animate_nouns_accusative>|
<masculine_inanimate_nouns_accusative>|
<femininee_nouns_accusative>| <plural_animate_nouns_accusative>|
<plural_inanimate_nouns_accusative>)|

<prepositions_ablative_of_time_place>
(<masculine_nouns_ablative>| <feminine_nouns_ablative>|
<neuter_nouns_ablative>| <plural_nouns_ablative>)|

<prepositions_prepositional_of_time_place>
(<masculine_nouns_prepositional>|
<feminine_nouns_prepositional>| <neuter_nouns_prepositional>|
<plural_nouns_prepositional>));
```

Рис. 6. Правила для обстоятельства.

```
<sentence_indirect_order_1p> = [<adverbial_group>] [<particles>]

(<subject_group_direct_order_first_person_masculine>|
<subject_group_indirect_order_1_first_person_masculine>|
<subject_group_indirect_order_2_first_person_masculine>|
<subject_group_direct_order_first_person_feminine>|
<subject_group_indirect_order_1_first_person_feminine>|
<subject_group_indirect_order_2_first_person_feminine>)

[<adverbial_group>] [<particles>]

<predicate_group_transitive_first_person>

[<adverbial_group>]

[([<particles>] <object_group_direct_order_genetive>| [<particles>]
<object_group_indirect_order_1_genetive>| [<particles>]
<object_group_indirect_order_2_genetive>| [<particles>]
<object_group_direct_order_accusative>| [<particles>]
<object_group_indirect_order_1_accusative>| [<particles>]
<object_group_indirect_order_2_accusative>)]

[<adverbial_group>];
```

Рис. 7. Правило для предложения с переходным глаголом с прямым порядком слов для 1 лица, единственного числа.

• *vocab_size* = *40000* – объем будущего словаря, количество компонентов, на которые токенизатор разобьет входной текст;

• *min_frequency* = 2 – минимальный порог частоты.

После получения файлов токенизатора, которые содержат объединенные токенизированные подстроки и их индексы, была определена конфигурация модели. Мы выбрали для предварительного обучения базовую модель RoBERTa, которая получила следующие параметры:

• *vocab_size* = 40000 – объем словаря, созданного токенизатором;

• *max_position_embeddings* = 512 – максимальная длина последовательности;

• *num_attention_heads* = 12 – количество головок внимания для каждого слоя внимания в кодировщике;

• *num_hidden_layers* = 12 – количество скрытых слоев в кодировщике;

• *type_vocab_size* = 1 – словарный запас.

Кроме того, была определена функция *DataCollatorForLanguageModeling()*, которая должна маскировать слова в данных с вероятностью 15%. В качестве алгоритма обучения использовался 8-разрядный Adam из библиотеки bitsandbytes[6], совмещающий в себе эффективность оригинального алгоритма Adam с более экономичным потреблением памяти за счет квантизованного (в 8 бит) представления истории градиента функции потерь по весам.

СПИСОК ЛИТЕРАТУРЫ

1. *Турдаков Д.Ю., Аветисян А.И., Архипенко К.В., Анциферова А.В., Ватолин Д.С., Волков С.С., Гасников А.В., Девяткин Д.А., Дробышевский М.Д., Коваленко А.П., Кривоносов М.И., Лукашевич Н.В., Малых В.А., Николенко С.И., Оселедец И.В., Перминов А.И., Сошенков И.В., Тихомиров М.М., Федотов А.Н., Хачай М.Ю.* Доверенный искусственный интеллект: вызовы и перспективные решения, 2022. doi:
2. *Shumailov I., Shumaylov Z., Kazhdan D., Zhao Y., Papernot N., Erdogdu M.A., Anderson R.* Manipulating SGD with Data Ordering Attacks, 2021. https://doi.org/10.48550/arXiv.2104.09667
3. *Kataoka H., Okayasu K., Matsumoto A., Yamagata E., Yamada R., Inoue N., Nakamura A., Satoh Y.* Pre-training without Natural Images, 2020. https://doi.org/10.48550/arXiv.2101.08515
4. *Liu Y., M Ott M., Goyal N., Du J., Joshi M., Chen D., Levy O., Lewis M., Zettlemoyer L., Stoyanov V.* RoBERTa: A Robustly Optimized BERT Pretraining Approach, 2019. https://doi.org/10.48550/arXiv.1907.11692
5. JSpeech Grammar Format https://www.w3.org/TR/2000/NOTE-jsgf-20000605/. Дата обращения: 2023-05-08.
6. *Баранов М.Т., Костяева Т.А., Прудникова А.В.* Под ред. Н.М. Шанского Русский язык: Справ. материалы: Учеб. пособие для учащихся. 4-е изд. – М.: Просвещение, 1988. 288 с.
7. *Лукашевич Н.В.* Тезаурусы в задачах информационного поиска. Изд-во Московского университета, 2011, 512 с.
8. *Shavrina T., Shapovalova O.* To the methodology of corpus construction for machine learning: "Taiga" syntax tree corpus and parser // Proceedings of the international conference "Corpus linguistics-2017", 2017. P. 78–84.
9. *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 // In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020. P. 4717–4726.
10. *Korobov M.* Morphological Analyzer and Generator for Russian and Ukrainian Languages // Analysis of Images, Social Networks and Texts, 2015. P. 320–332.
11. *Straka M., Straková J.* Tokenizing. POS Tagging, Lemmatizing and Parsing UD 2.0 with UDPipe, 2017. https://doi.org/10.18653/v1/K17-3009
12. *Honnibal M., Montani I.* spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing, 2017.
13. JSGFTools: Some tools for JSGF grammar expansion [https://github.com/syntactic/JSGFTools]. Дата обращения: 2023-05-10.

[6] https://github.com/TimDettmers/bitsandbytes

SAFE PRE-TRAINING OF DEEP LANGUAGE MODELS IN A SYNTHETIC PSEUDO-LANGUAGE

T. E. Gorbacheva[a] and I. Bondarenko[a]

[a] *Novosibirsk State University, Novosibirsk, Russian Federation*

Presented by Academician of the RAS A.L. Semenov

This paper compares the pre-training of a transformer on natural language texts and on sentences of a synthetic pseudolanguage. The artificial texts were automatically generated according to the rules we wrote in a context-free grammar. The results of fine-tuning to complete tasks of the RussianSuperGLUE project statistically reliably showed that the models had the same scores. That is, we can consider that the use of artificial texts provides an advantage in the AI safety due to the ability to completely control the composition of the dataset. We can also say that at the pre-training stage of a model like RoBERTa, it is enough to learn to recognize only the syntactic and morphological patterns of the language, which can be successfully created in a fairly simple way, such as a context-free grammar.

Keywords: deep learning methods, transformers, pre-training, automatic text generation, language models, synthetic data, AI safety