УДК *004.8*

Д. В. Брылев

Иркутский национальный исследовательский технический университет, г. Иркутск, Российская Федерация

ИССЛЕДОВАНИЕ ПРЕДОБУЧЕННЫХ МОДЕЛЕЙ НЕЙРОННЫХ СЕТЕЙ ДЛЯ ГЕНЕРАЦИИ ТЕКСТА

Аннотация. *Генерация текста в настоящее время становится одной из самых популярных технологий машинного обучения. Предобученные модели позволяют: генерировать текст в качестве ответа на вопрос, на основе имеющегося текста генерировать последующее в нем слово, генерировать осмысленный текст для каналов коммуникации и др. Основная информация по архитектурам моделей нейронных сетей для генерации текста представлена преимущественно в англоязычных источниках, в русскоязычной литературе отсутствуют обзорные статьи по этой теме, в связи с этим теоретические данные являются разрозненными. В данной статье представлен обзор существующих современных моделей нейронных сетей для генерации текста. Рассматривается архитектура нейронной сети «трансформер»: принцип работы данной архитектуры, описываются категории нейросетевых моделей «трансформер» и приводятся примеры решаемых ими задач. Рассматриваются методы генерации текста: жадная генерация (greedy search), лучевой поиск (beam search), сэмплирование с температурой и сэмплирование с ограничением маловероятных токенов. Проведено сравнение предобученных моделей нейронных сетей для генерации текста. Исследование предобученных моделей нейронных сетей для генерации текста позволит определить, какие модели являются более предпочтительными для определенной задачи по генерации текста. Проведение данного исследования поможет в дальнейшем определиться с выбором наиболее подходящей предобученной модели нейронной сети для поставленной задачи, связанной с генерацией текста для каналов коммуникации.*

Ключевые слова: *генерация текста, нейронные сети, архитектура «трансформер», предобученная модель, токен.*

D. V. Brylev

Irkutsk National Research Technical University, Irkutsk, the Russian Federation

RESEARCH OF PRE-TAINED NEURAL NETWORKS MODELS FOR TEXT GENERATION

Abstract. *Text generation is currently becoming one of the most popular machine learning technologies. Pre-trained models allow: to generate text as an answer to a question, to generate a subsequent word on the basis of an existing text, to generate meaningful text for communication channels, etc. The main information on the architecture of neural network models for text generation is presented mainly in English-language sources, in Russian-language literature there are no review articles on this topic, in this regard, the theoretical data are scattered. This article presents a review of existing modern models of neural networks for text generation. The Transformer architecture of neural network is considered: the principle of operation of this architecture, categories of Transformer neural network models are described and examples of problems solved by them are given. Text generation methods are considered: greedy search, beam search, temperature sampling and sampling with low-probability tokens limitation. A comparison of pre-trained neural network models for text generation was carried out. The study of pre-trained neural network models for text generation will help in determining which models are more preferable for a particular text generation task. Conducting this study will help in the future to determine the choice of the most appropriate pre-trained neural network model for the task of generating text for communication channels.*

Keywords: *text generation, neural networks, Transformer architecture, pre-trained model, token.*

Введение

В настоящее время искусственный интеллект стал неотъемлемой частью современных технологий и находит применение во многих областях нашей жизни. Одной из наиболее популярных областей для использования нейросетей является генерация текста. С появлением нейросетевой архитектуры «трансформер» качество генерации текстов естественного языка значительно улучшилось [1-3]. Нейронные сети способны генерировать тексты в любой предметной области. Создание текста является сложной проблемой обработки естественного

языка. В настоящее время в мире имеется большое количество различных нейросетевых моделей, способных генерировать текст, но в русскоязычной литературе отсутствует полный обзор по существующим современным моделям нейронных сетей для генерации текста.

Нейросетевая архитектура «трансформер»

Основными моделями для генерации текста являются модели с архитектурой «трансформер» [1-3]. Впервые модель Transformer была предложена исследователями из Google Research и Google Brain в работе 2017 года «Attention Is All You Need» [1].

Исходная архитектура «трансформер» состоит из двух блоков:

1) кодировщика, который получает входные данные (например, последовательность слов) и создает их представление (набор чисел);

2) декодировщика, который использует представление, полученное на выходе из кодировщика, для генерации новых данных (например, слов, переведенных на другой язык) [1, 4].

Каждый из блоков содержит по несколько слоев (наборов математических операций), в оригинальной модели «трансформер» их шесть, но это число можно изменять.

Схема оригинальной архитектуры «трансформер» представлена на рис. 1 [1].

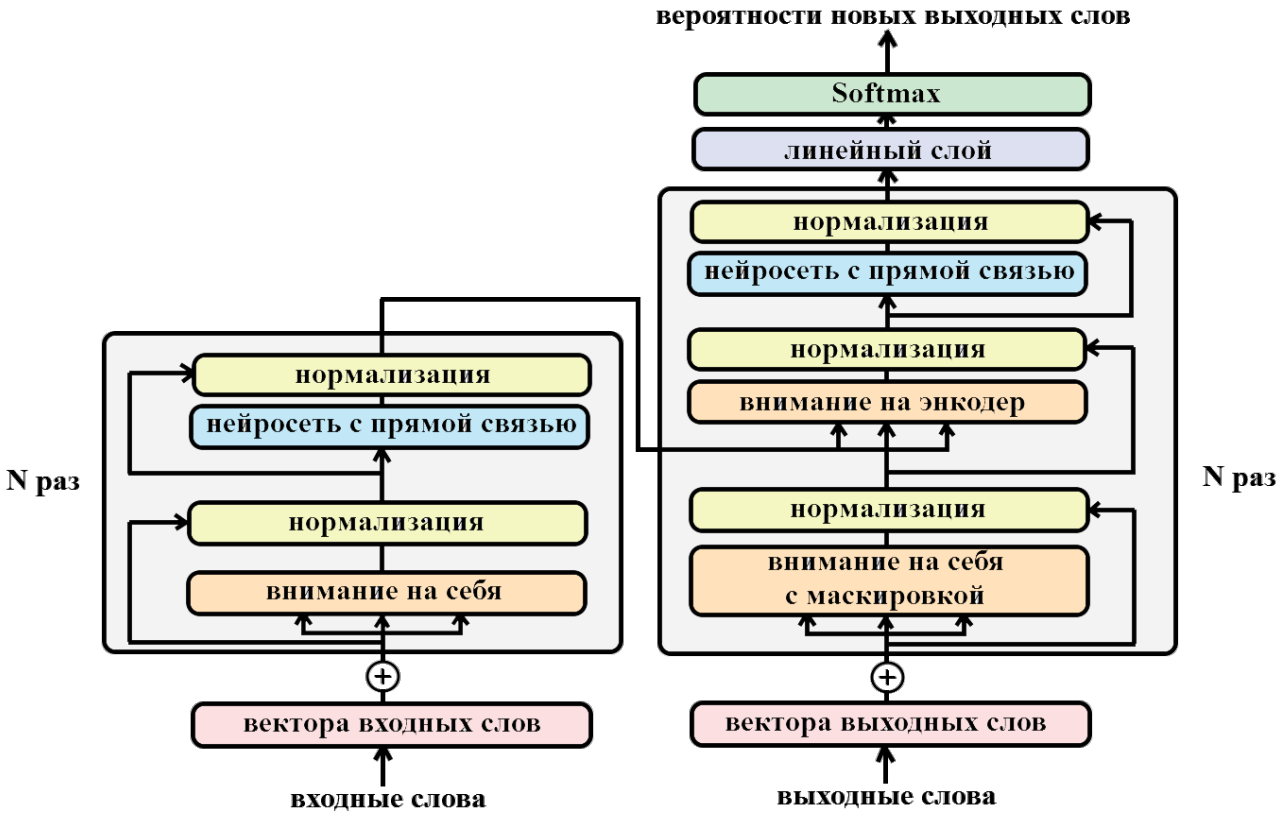

Рис. 1. Архитектура оригинальной модели Transformer (кодировщик слева, декодировщик – справа) [1]

Принцип работы архитектуры «трансформер» заключается в следующем:

1) кодировщик кодирует стопку (батч) предложений за раз, пропуская эту стопку через шесть своих слоев. Все сопровождается нормализацией, она обязательна и в декодировщике. Кодирование завершено [1, 5];

2) после работы кодировщика первый слой декодировщика начинает работу с матрицей уже сгенерированных слов: пока ничего не сгенерировано, все значения входной матрицы декодировщика «замаскированы». Ничто не получает веса внимания, декодировщик включает «внимание на кодировщик»;

3) с последнего слоя декодировщика результат попадает на «финальный слой», где вектор превращается в слово. Первый временной шаг декодирования закончен;

4) на втором временном шаге слой «внимание на себя» смотрит на обновленную входную матрицу для декодировщика, где есть первое выданное нейросетью слово. Затем нейросеть выполняет слой за слоем, и декодировщик выдает второе слово;

5) на третьем шаге во входной матрице декодировщика уже два прошлых слова, после этого можно декодировать слова до тех пор, пока входная матрица декодировщика не заполнится до конца и не сгенерируется сигнал остановки [1].

Языковые модели могут обрабатывать только числа, поэтому необходимо преобразовывать вводимые нами текстовые данные в числовые данные. Осуществляется это при помощи метода токенизации. Текст разделяется на слова (или части слов, символы и др.), обычно называемые токенами, а затем эти токены преобразуются в числа, чтобы мы могли построить из них тензор и передать их в модель. Числовое обозначение токена называют «эмбеддингом». Каждая модель может обработать лишь определенное число токенов [6].

Существенным ограничением является тот факт, что для обучения текстовой модели, особенно большой, требуется огромный объем данных. Это очень затратно с точки зрения времени и вычислительных ресурсов и даже приводит к воздействию на окружающую среду.

Если бы каждый раз исследовательская группа, студенческая организация или компания обучали модель с нуля – это привело бы к большим глобальным затратам [7].

Поэтому использование предобученных языковых моделей имеет первостепенное значение: построение на основе уже обученных весов снижает общую стоимость вычислений и углеродный след сообщества.

Последовавшие за «Attention Is All You Need» исследования показали, что каждый из блоков «трансформера» можно использовать независимо друг от друга. Этот подход предполагает использование одного из блоков «трансформера»: кодировщика или декодировщика, его обучение на огромных объемах текстовых данных и выполнение обширных вычислений на нем [8-10].

Модели «трансформеры» можно разделить на три категории:

1) подобные GPT;

2) BERT-подобные;

3) подобные T5.

Упрощенное схематическое изображение нейросетевых моделей архитектуры «трансформер» представлено на рис. 2 [8-10].

GPT (генеративный предобученный «трансформер») – семейство языковых моделей для генерации (продолжения) текста, впервые представленное OpenAI в 2018 году. GPT используют только декодировщик модели «трансформер». На каждом этапе для данного слова слои внимания могут получить доступ только к словам, расположенным перед ним в предложении. Предварительное обучение моделей декодировщиков обычно основывается на предсказании следующего слова в предложении.

Для того, чтобы понимать и анализировать слова, модель использует матрицу эмбеддингов – важный компонент ее архитектуры. Матрица эмбеддингов состоит из строк числовых наборов, которые представляют слова и отражают определенные аспекты их значения. Эти наборы могут различаться по размеру в зависимости от размеров модели GPT.

Чтобы обеспечить всестороннее понимание слов, важно учитывать их значение в контексте. За это отвечают слои внутреннего внимания. Внутреннее внимание позволяет модели идентифицировать подходящие и взаимосвязанные слова, облегчая оценку контекстных подсказок для каждого слова перед дальнейшей обработкой в нейронной сети. Для этого каждому слову присваиваются коэффициенты, обозначающие его релевантность в данном сегменте текста. Эти коэффициенты затем включаются в векторы представления соответствующих слов [8].

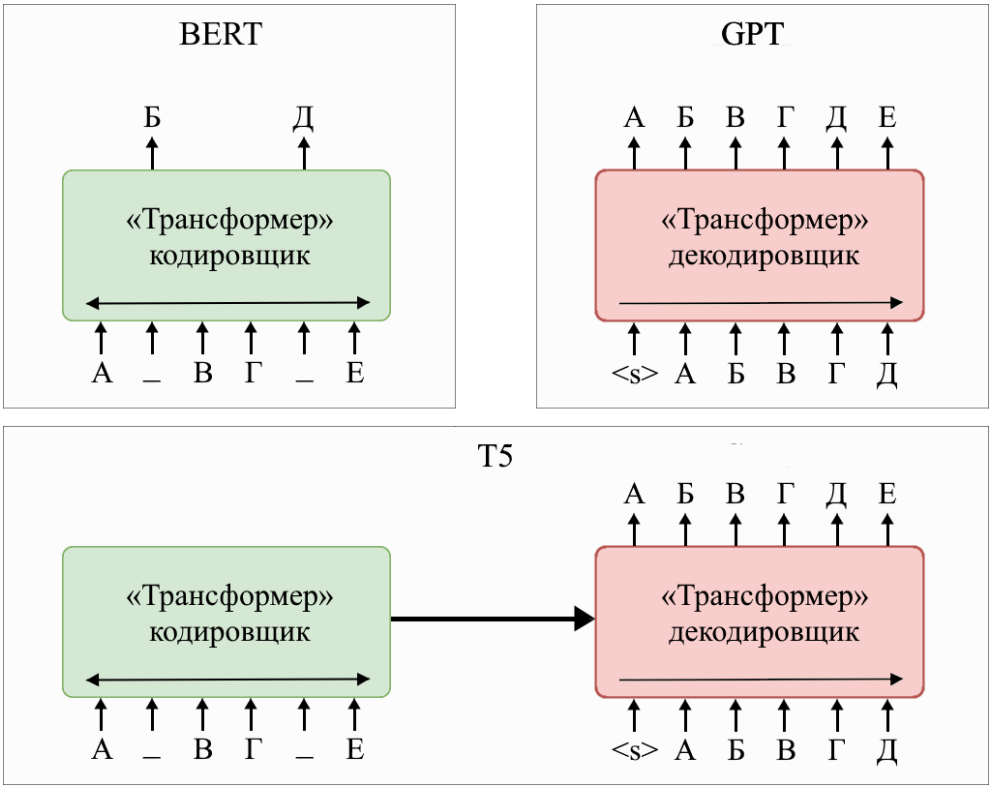

Рис. 2. Упрощенное схематическое изображение моделей Transformer [8-10]

Модель двунаправленного «трансформера» кодировщика – BERT (Bidirectional Encoder Representations from Transformers) была предложена исследователями из Google в работе 2018 года «BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding» [9]. Что отличает BERT, так это его уникальный подход к предварительному обучению. В отличие от предыдущих моделей, предсказывающих следующее слово в предложении, BERT использует маскированное языковое моделирование. Модель предобучается, предсказывая замаскированное слово в предложении путем совместной обработки левого и правого контекста на всех слоях нейросети. Например, предсказывая слово в начале предложения, в котором искомое слово заменено токеном [MASK] (рис. 3). Этот метод позволяет BERT рассматривать всю фразу, а не только предыдущие или последующие слова [9].

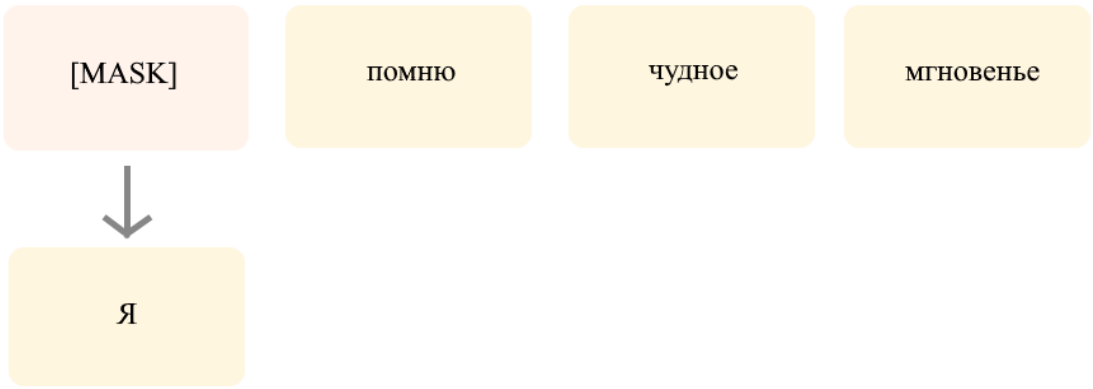

Рис. 3. Пример моделирования маскированного слова [9]

T5 (text-to-text transfer transformer) – нейросетевая модель, использующая всю архитектуру «трасформер», представлена исследователями из Google в работе 2019 года «Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer» [10]. T5 использует подход преобразования текста в текст. Каждая задача, включая перевод, ответы на вопросы и классификацию, представляет собой подачу текста в качестве входных данных на блок коди-

ровщика (подобно BERT) и обучение ее генерации некоторого целевого текста при помощи блока декодировщика (подобно GPT) [11].

Основная идея T5 заключается в том, что сначала выполняется процесс предварительного обучения модели на некотором отдельном наборе данных с постановкой задачи для самообучения (например, заполнение пропущенных слов в предложениях), а затем точная настройка (или дальнейшее обучение) этой модели на последующем целевом наборе данных, связанных с конкретной задачей (например, перевод или обобщение текста) [10].

Также T5 применим для решения задачи размерного заполнения пропусков (sized fill-in-the-blank). В данной задаче модели предлагается заменить пропуск указанным количеством слов, например, если мы предоставим модели входные данные «Мне нравятся книги _5_ жанров», модель будет обучена заполнять пропуск пятью словами. Этот подход расширяет возможности модели генерировать связный и содержательный текст на основе заданной подсказки.

Модели кодировщики используют только кодировщик модели «трансформер». На каждом этапе слои внимания могут получить доступ ко всем словам в исходном предложении. Эти модели часто характеризуются как имеющие «двунаправленное» внимание и часто называются моделями автоматического кодирования.

Предварительное обучение этих моделей обычно основывается на том, чтобы как-то исказить данное предложение (например, путем маскировки в нем случайных слов) и поставить перед моделью задачу найти или восстановить исходное предложение [9, 12].

Модели кодировщика лучше всего подходят для задач, требующих понимания всего предложения, таких как классификация предложений, распознавание именованных сущностей (и, в более общем плане, классификация слов) и ответы на вопросы с извлечением [9].

Модели декодировщики используют только декодировщик модели «трансформер». На каждом этапе для данного слова слои внимания могут получить доступ только к словам, расположенным перед ним в предложении. Эти модели часто называют авторегрессионными моделями.

Предварительное обучение моделей декодировщиков обычно основывается на предсказании следующего слова в предложении.

Эти модели лучше всего подходят для задач, связанных с генерацией текста [8, 12].

Модели кодировщик-декодировщик (также называемые моделями «sequence-to-sequence») используют обе части архитектуры «трансформер». На каждом этапе уровни внимания кодировщика могут получить доступ ко всем словам в исходном предложении, тогда как уровни внимания декодировщика могут получить доступ только к словам, расположенным перед данным словом во входных данных.

Предварительное обучение этих моделей может быть выполнено с использованием целей моделей кодировщика или декодировщика, но обычно включает в себя что-то более сложное. Например, T5 предварительно обучается путем замены случайных фрагментов текста (которые могут содержать несколько слов) одним специальным словом маски, а затем цель состоит в том, чтобы предсказать текст, который заменяет это слово маски.

Модели кодировщик-декодировщик лучше всего подходят для задач, связанных с созданием новых предложений в зависимости от заданных входных данных, таких как обобщение, перевод или генеративный ответ на вопрос [10, 12].

Модели архитектуры «трансформер» и примеры решаемых ими задач представлены в таблице 1.

Модели можно разделить на большие, обучение которых занимало месяца, и малые, обученные на меньшем объеме данных.

Поскольку более легковесные модели имеют меньшее количество параметров, чем оригинальные, они могут быть использованы на менее мощных вычислительных устройствах, в том числе и на мобильных устройствах и бесплатных средах облачных вычислений, что делает их более доступными для дальнейшего дообучения под конкретные задачи [13, 14].

Сравнение моделей архитектуры «трансформер» по количеству параметров представлено в таблице 2 [15].

Таблица 1

Модели архитектуры «трансформер» и примеры решаемых ими задач

Архитектура «трансформер»	Модель	Решаемые задачи
Кодировщик	BERT	1) классификация предложений; 2) распознавание именованных сущностей; 3) ответы на вопросы с извлечением
Декодировщик	GPT	генерация текста
Кодировщик -декодировщик	T5	1) обобщение; 2) перевод; 3) генеративный ответ на вопрос

Таблица 2

Сравнение моделей архитектуры «трансформер» по количеству параметров

Семейство моделей	Название модели	Количество параметров модели (в миллионах)
GPT	GPT-4	500 000
	ruGPT-3 XL	1 300
	ruGPT-3 Large	760
	ruGPT-3-medium	356
BERT	ruBERT large	427
	RuBERT (DeepPavlov)	180
	ruBERT-base	178
	BERT-Base	110
T5	FRED-T5 1.7B	1 740
	mT5-large (Google)	973
	FRED-T5 large	820
	ruT5-large	737
	mT5-base (Google)	390
	ruT5-base	222

Методы генерации текста

Языковая модель выдает распределение вероятностей следующего токена, а эту информацию можно по-разному использовать для генерации текста.

Самый простой метод декодирования – это жадная генерация (greedy search), когда мы каждый раз выбираем токен с наибольшей вероятностью в качестве продолжения текста. Основным недостатком жадной генерации является то, что она пропускает слова с высокой ве-

роятностью, которые скрыты за словами с низкой вероятностью, как это видно на рис. 4: слово «была» с высокой вероятностью 0.9 скрывается за словом «собака», поэтому жадная генерация пропускает последовательность слов: «Эта», «собака», «была» [8, 16].

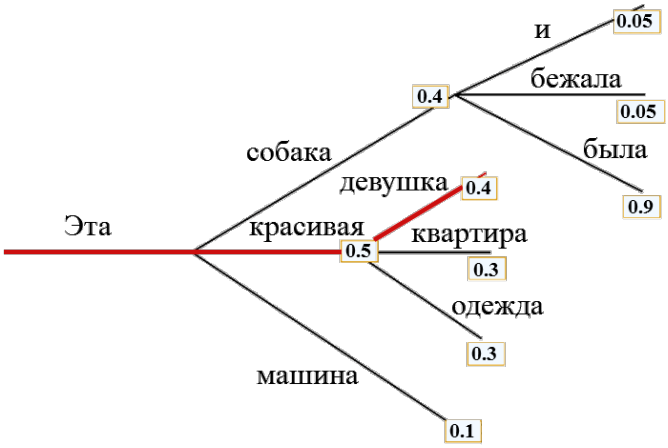

Рис. 4. Пример жадной генерации (greedy search) [16]

Другим методом является лучевой поиск (beam search), который снижает риск пропуска скрытых последовательностей слов с высокой вероятностью, сохраняя наиболее вероятные гипотезы на каждом временном шаге и в конечном итоге выбирая гипотезу с наибольшей общей вероятностью (рис. 5) [16]. Вместо того, чтобы просто выбирать наиболее вероятный токен на каждом этапе, мы расширяем нашу выборку, рассматривая одновременно заданное количество токенов (размер луча). Это позволяет нам исследовать несколько путей в процессе генерации, что приводит к более широкому выбору вариантов сгенерированного текста. В итоге мы можем оценить эти варианты на основе их перплексии и выбрать наиболее подходящий вариант, соответствующий желаемому результату. Такая генерация обладает хорошей когерентностью, но обычно у нее не хватает «человечности». Как утверждается в работе «The Curious Case of Neural Text Degeneration», высококачественный человеческий язык не следует распределению следующих слов с высокой вероятностью [17]. Другими словами, как люди, мы хотим, чтобы сгенерированный текст удивлял нас, а не был скучным или предсказуемым.

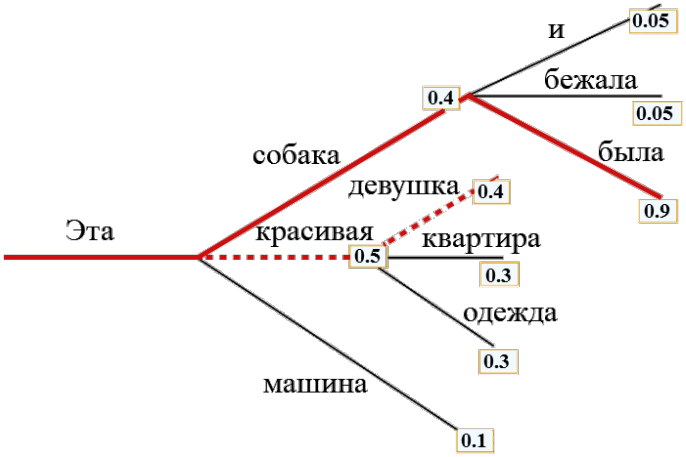

Рис. 5. Пример менее жадной генерации (beam search) [16]

Сэмплирование с температурой. Чтобы добавить тексту непредсказуемости и человечности можно использовать вероятностное сэмплирование с температурой. При такой генерации мы берем не самый вероятный токен, а выбираем его «случайно» с учетом распределения вероятностей. Параметр температуры позволяет контролировать степень хаотичности. Используя коэффициент temperature происходит увеличение вероятности использования

слов с высокими значениями вероятности и уменьшение вероятности использования слов с низкой вероятностью в распределении. Также, чем ближе к нулю значение, тем больше генерация будет похожа на жадный подбор слов. То есть мы будем больше подвержены формату того, что было в обучающей выборке.

Сэмплирование с ограничением маловероятных токенов. В методе top-k выбираются k наиболее вероятных следующих слов, и вероятности распределяется только между этими k следующими словами. Вместо выбора только из k наиболее вероятных слов, в методе top-p выборка строится из наименьшего возможного набора слов, совокупная вероятность которых превышает вероятность p. Затем вероятности распределяется среди этого набора слов. Таким образом, размер набора слов (количество слов в наборе) может динамически увеличиваться и уменьшаться в зависимости от распределения вероятности следующего слова [8, 16].

Заключение

В зависимости от того, какую задачу по генерации текста необходимо решить, нужно использовать подходящую для этого предобученную модель: использующую всю архитектуру «трансформер» или только кодировщик или декодировщик.

Для задач генерации связного текста используются GPT-подобные «трансформеры» декодировщики, генерация предложений происходит на основе выборки из языковой модели, которая дает распределение вероятностей следующего слова с учетом предыдущих контекстов. BERT-подобные модели кодировщики не подходят для решения подобных задач ввиду своей двунаправленной природы, однако могут применяться для генерации пропущенных в тексте слов. Также «трансформеры» кодировщики можно использовать для извлечения ответов на вопросы, модель принимает отрывок текста и вопрос в качестве входных данных, а затем возвращает фрагмент отрывка, который с наибольшей вероятностью отвечает на вопрос. Подобные T5 модели подходят для задач, связанных с созданием новых предложений в зависимости от заданных входных данных, таких как конспектирование или генерация текста в качестве ответа на вопрос [8-10].

Качество генерации зависит от количества параметров модели, однако, чем меньше параметров у модели, тем меньше вычислительных ресурсов требуется для ее использования. На моделях с малым числом параметров наблюдаются проблемы с качеством генерации текста. Учитывая большое количество готовых полных моделей, следует изучить, какие результаты могут показать модели, дообученные под конкретную задачу.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., Polosukhin, I. Attention is all you need // Advances in Neural Information Processing Systems 30. 2017. pp. 5998-6008.
2. Yang Z., Keung J., Yu X., Gu X., Wei Z., Ma X., Zhang M. A Multi-Modal Transformer-based Code Summarization Approach for Smart Contracts // The 2021 International Conference on Program Comprehension. 2021. pp. 1-12.
3. Juraska J., Walker M. Attention Is Indeed All You Need: Semantically Attention-Guided Decoding for Data-to-Text NLG // Proceedings of the 14th International Conference on Natural Language Generation. 2021. pp. 416-431.
4. The Illustrated Transformer [Электронный ресурс]. – Режим доступа. – URL: http://jalammar.github.io/illustrated-transformer (дата обращения: 10.04.2023).
5. Lei Ba J., Kiros J.R., Hinton G.E. Layer Normalization. ArXiv. 2016. URL: https://arxiv.org/pdf/1607.06450.pdf (дата обращения: 11.04.2023).
6. Как устроена нейросеть BERT от Google [Электронный ресурс]. – Режим доступа. – URL: https://sysblok.ru/knowhow/kak-ustroena-nejroset-bert-ot-google (дата обращения: 17.04.2023).
7. Eco2AI: контроль углеродного следа моделей машинного обучения в качестве первого шага к устойчивому искусственному интеллекту / С. А. Буденный, В. Д. Лазарев, Н. Н. Захаренко

[и др.] // Доклады Российской академии наук. Математика, информатика, процессы управления. – 2022. – Т. 508, № 1. – С. 134-145.

8. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language Models are Unsupervised Multitask Learners.

9. Devlin J., Chang M., Lee K., Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding // Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). 2019. pp. 4171–4186.

10. 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 // Journal of Machine Learning Research, Volume 21. 2020. pp. 1-67.

11. Многозадачная модель T5 для русского языка [Электронный ресурс]. – Режим доступа. – URL: https://habr.com/ru/articles/581932 (дата обращения: 19.04.2023).

12. Васюнин, М. А. Технологии понимания естественного языка / М. А. Васюнин, А. А. Бахман // Искусственный интеллект в автоматизированных системах управления и обработки данных: Сборник статей Всероссийской научной конференции. В 2-х томах, Москва, 27–28 апреля 2022 года. Том 2. – Москва: Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет), 2022. – С. 269-274.

13. Sanh V., Debut L., Chaumond J., Wolf T. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. ArXiv. 2019. URL: https://arxiv.org/abs/1910.01108 (дата обращения: 20.04.2023).

14. Hahn S., Choi H. Self-Knowledge Distillation in Natural Language Processing // Proceedings of the International Conference on Recent Advances in Natural Language Processing, Varna, Bulgaria, September 2-4, 2019. 2019. pp. 423-430.

15. Галеев, Д. Т. Экспериментальное исследование языковых моделей «трансформер» в задаче нахождения ответа на вопрос в русскоязычном тексте / Д. Т. Галеев, В. С. Панищев // Информатика и автоматизация. – 2022. – Т. 21, № 3. – С. 521-542.

16. How to generate text: using different decoding methods for language generation with Transformers [Электронный ресурс]. – Режим доступа. – URL: https://huggingface.co/blog/how-to-generate (дата обращения: 20.04.2023).

17. Holtzman A., Buys J., Du L., Forbes M., Choi Y. The Curious Case of Neural Text Degeneration. ArXiv. 2019. URL: https://arxiv.org/abs/1904.09751 (дата обращения: 20.04.2023).

REFERENCES

1. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., Polosukhin, I. Attention is all you need // Advances in Neural Information Processing Systems 30. 2017. pp. 5998-6008.

2. Yang Z., Keung J., Yu X., Gu X., Wei Z., Ma X., Zhang M. A Multi-Modal Transformer-based Code Summarization Approach for Smart Contracts // The 2021 International Conference on Program Comprehension. 2021. pp. 1-12.

3. Juraska J., Walker M. Attention Is Indeed All You Need: Semantically Attention-Guided Decoding for Data-to-Text NLG // Proceedings of the 14th International Conference on Natural Language Generation. 2021. pp. 416-431.

4. The Illustrated Transformer [Electronic resource]. – Access mode. – URL: http://jalammar.github.io/illustrated-transformer (accessed: 10.04.2023).

5. Lei Ba J., Kiros J.R., Hinton G.E. Layer Normalization. ArXiv. 2016. URL: https://arxiv.org/pdf/1607.06450.pdf (accessed: 11.04.2023).

6. How Google's BERT neural network works [Electronic resource]. – Access mode. – URL: https://sysblok.ru/knowhow/kak-ustroena-nejroset-bert-ot-google (accessed: 17.04.2023).

7. Eco2AI: Controlling the carbon footprint of machine learning models as the first step towards sustainable artificial intelligence / Budyonny S.A., Lazarev V.D., Zakharenko N.N. [et al.] // Reports of

the Russian Academy of Sciences. Mathematics, informatics, control processes. – 2022. – V. 508, № 1. – pp. 134-145.

8. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language Models are Unsupervised Multitask Learners.

9. Devlin J., Chang M., Lee K., Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding // Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). 2019. pp. 4171–4186.

10. 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 // Journal of Machine Learning Research, Volume 21. 2020. pp. 1-67.

11. Multitasking model T5 for the Russian language [Electronic resource]. – Access mode. – URL: https://habr.com/ru/articles/581932 (accessed: 19.04.2023).

12. Vasyunin, M. A. Technologies for understanding natural language / M. A. Vasyunin, A. A. Bakhman // Artificial intelligence in automated control systems and data processing: Collection of articles of the All-Russian Scientific Conference. In 2 volumes, Moscow, April 27–28, 2022. Volume 2. – Moscow: Moscow State Technical University named after N.E. Bauman (National Research University), 2022. – pp. 269-274.

13. Sanh V., Debut L., Chaumond J., Wolf T. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. ArXiv. 2019. URL: https://arxiv.org/abs/1910.01108 (accessed: 20.04.2023).

14. Hahn S., Choi H. Self-Knowledge Distillation in Natural Language Processing // Proceedings of the International Conference on Recent Advances in Natural Language Processing, Varna, Bulgaria, September 2-4, 2019. 2019. pp. 423-430.

15. Galeev D., Panishchev V. Experimental Study of Language Models of «Transformer» in the Problem of Finding the Answer to a Question in a Russian-Language Text. / D. T. Galeev, V. S. Panishchev // Computer Science and Automation. – 2022. – V. 21, № 3. – pp. 521-542.

16. How to generate text: using different decoding methods for language generation with Transformers [Electronic resource]. – Access mode. – URL: https://huggingface.co/blog/how-to-generate (accessed: 20.04.2023).

17. Holtzman A., Buys J., Du L., Forbes M., Choi Y. The Curious Case of Neural Text Degeneration. ArXiv. 2019. URL: https://arxiv.org/abs/1904.09751 (accessed: 20.04.2023).

Информация об авторах

Брылев Данил Викторович – магистрант, Институт информационных технологий и анализа данных, Иркутский национальный исследовательский технический университет, г. Иркутск, e-mail: danil.brylev@mail.ru

Information about the authors

Brylev Danil Viktorovich – Master's student, Institute of Information Technology and Data Science, Irkutsk National Research Technical University, Irkutsk, e-mail: danil.brylev@mail.ru