УДК 004.827

СРАВНИТЕЛЬНЫЙ АНАЛИЗ МОДЕЛЕЙ ТРАНСФОРМЕРА ДЛЯ КЛАССИФИКАЦИИ НЕСТРУКТУРИРОВАННОЙ ТЕКСТОВОЙ ИНФОРМАЦИИ

Нежников Ратибор Игоревич, Астраханский государственный университет имени В. Н. Татищева, 414056, Российская Федерация, г. Астрахань, ул. Татищева, 20а,

аспирант, ORCID: 0009-0000-2042-4028, e-mail: nezhnikov1998@gmail.com

Марьенков Александр Николаевич, Астраханский государственный университет имени В. Н. Татищева, 414056, Российская Федерация, г. Астрахань, ул. Татищева, 20а,

кандидат технических наук, ORCID: 0000-0003-1378-3553, e-mail: marenkovan17@gmail.com

В данной статье представлен обзор подхода к классификации неструктурированной текстовой информации с использованием модели трансформера. Трансформеры, такие как BERT, GPT и RoBERTa, предлагают существенные преимущества в обработке текста и анализе данных благодаря своей мощной архитектуре и возможности к файн-тюнингу. Описываются основные этапы использования трансформеров в задаче классификации текстов, включая предобработку данных, выбор и настройку архитектуры трансформера, а также методы оценки производительности. В статье описан обзор проблемы классификации текстовой информации, основных методов ее решения, а также преимущества использования трансформеров в этой области. Приводится краткий обзор основных компонентов архитектуры трансформера, таких как механизм позиционного кодирования, многослойные перцептроны и механизм внимания. Анализируются методы подготовки данных для классификации, настройки и файн-тюнинга модели трансформера, а также методы оценки производительности классификации с использованием таких показателей, как точность, полнота и F1-мера. Приводятся практические результаты исследования нейросетевых моделей для решения задачи классификации, а также приведен анализ архитектур, размерности модели, регуляризации, оптимизатора, скорости обучения и размера батча, которые могут влиять на эффективность использования трансформеров для классификации текстов.

Ключевые слова: модель трансформера, BERT, RoBERTa, классификации текстовой информации, нейронные сети

COMPARATIVE ANALYSIS OF TRANSFORMER MODELS FOR CLASSIFICATION OF UNSTRUCTURED TEXT INFORMATION

Nezhnikov Ratibor I., Astrakhan Tatishchev State University, 20a Tatishchev St., Astrakhan, 414056, Russian Federation,

graduate student, ORCID: 0009-0000-2042-4028, e-mail: nezhnikov1998@gmail.com

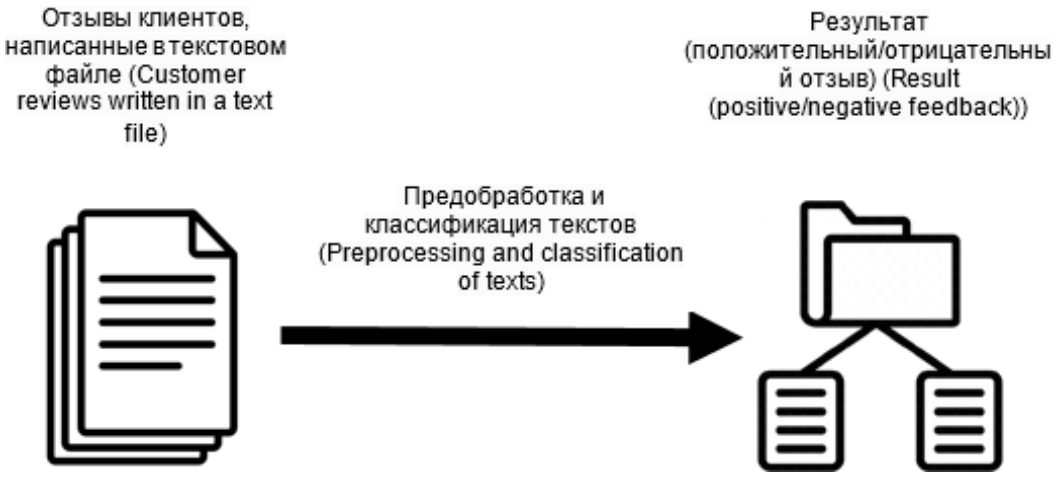
Marenkov Alexander N., Astrakhan Tatishchev State University, 20a Tatishchev St., Astrakhan, 414056, Russian Federation,

Cand. Sci. (Engineering), Associate Professor, ORCID: 0000-0003-1378-3553, e-mail: marenkovan17@gmail.com

This article presents an overview of an approach to classifying unstructured textual information using the transformer model. Transformers such as BERT, GPT and RoBERTa offer significant advantages in text processing and data analysis due to their powerful architecture and fine-tuning capabilities. The main stages of using transformers in the task of text classification are described, including data preprocessing, selection and configuration of the transformer architecture, as well as methods for evaluating performance. The article describes an overview of the problem of classifying text information, the main methods for solving it, as well as the advantages of using transformers in this area. Transformer Architecture Overview: Provides a brief overview of the major components of the Transformer architecture, including the attention engine, multilayer perceptrons, and positional encoding engine. Text classification using the Transformer model: Methods for preparing data for classification, tuning and fine-tuning the Transformer model are described, as well as approaches to assessing classification performance using metrics such as accuracy, precision, recall and F1-measure. Practical aspects: Discusses key factors that can influence the effectiveness of using transformers for text classification, such as choice of architecture, model dimension, regularization, optimizer, learning rate and batch size.

Keywords: transformer model, BERT, RoBERTa, text information classification, neural networks

Graphical annotation (Графическая аннотация)



ВВЕДЕНИЕ

Обработка и анализ текстовых данных являются одними из наиболее важных задач в современном мире. С каждым днем количество неструктурированной текстовой информации увеличивается благодаря появлению интернета, социальных сетей, блогов и других цифровых платформ. В результате необходимы мощные инструменты для анализа и классификации текстов, которые могут быть применены в широком спектре областей, таких как маркетинг, журналистика, научные исследования, правоохранительные органы и т. д. [1].

За последнее время машинное обучение и искусственный интеллект стали основными инструментами для работы с текстовыми данными. Идея архитектуры трансформера, предложенная Vaswani и его коллегами в 2017 г. [11], является одним из важных прорывов в этой области. В работах [10, 12, 14] при решении задач обработки естественного языка (NLP), таких как машинный перевод, извлечение информации, генерация текста, определение тональности текста и т. д., архитектуры BERT, GPT и RoBERTa имеют низкий процент ошибок.

Целью данной статьи является исследование применения трансформеров для классификации неструктурированной текстовой информации. Для этого необходимо рассмотреть их основные принципы работы, архитектуру, а также способы применения для решения задач классификации. В рамках экспериментальной части необходимо обучить модель на реальном наборе данных и проанализировать полученные результаты с использованием различных метрик производительности. По результатам проведенного исследования можно будет оценить эффективность моделей и определить возможность для дальнейшего применения в задаче классификации текстов.

ТРАНСФОРМЕРЫ В ОБРАБОТКЕ ЕСТЕСТВЕННОГО ЯЗЫКА

Трансформер – это архитектура глубокого обучения. Авторы архитектуры революционизировали обработку естественного языка, и их разработка стала основой для создания множества моделей, таких как BERT, GPT-3, T5 и т. д.

Основными преимуществами трансформеров являются их способность к параллельной обработке данных и механизм внимания (attention), позволяющий модели эффективно обрабатывать информацию из разных частей текста [12].

Механизм внимания вычисляет взвешенную сумму значений (values) на основе сходства между запросами (queries) и ключами (keys). В контексте трансформера запросы, ключи и значения генерируются из векторных представлений слов в тексте. Формула для механизма внимания следующая:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \quad (1)$$

где Q – матрица запросов;

K – матрица ключей;

V – матрица значений;

d_k – размерность ключей и запросов.

Важным элементом трансформера является механизм самовнимания (self-attention), который вычисляет взаимосвязи между словами в тексте. Самовнимание является частным случаем механизма внимания, где запросы, ключи и значения берутся из одного и того же источника (текущего входа или предыдущего слоя трансформера) [8].

Трансформеры используют многоголовое внимание для улучшения обработки информации с разных позиций и представлений. Многоголовое внимание состоит из нескольких параллельных механизмов внимания, называемых «головами». Результаты каждой головы объединяются и передаются на следующий слой. Формула для многоголового внимания:

$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head}_1, \dots, \mathrm{head}_h)W^O, \quad (2)$$

где $\mathrm{head}_i = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$;

W_i^Q, W_i^K, W_i^V – весовые матрицы для i-й головы [37].

ПОЗИЦИОННАЯ КОДИРОВКА (POSITIONAL ENCODING)

Трансформеры не имеют встроенной позиционной информации, поскольку они не используют рекуррентные или сверточные слои. Для добавления позиционной информации к входным векторам слов используется позиционная кодировка. Она добавляется к входным векторам перед подачей их на слой самовнимания.

Позиционная кодировка может быть статической или динамической. В оригинальной статье о трансформерах используется статическая позиционная кодировка, которая вычисляется по формуле:

$$\mathrm{PE}(\mathrm{pos}, \mathrm{i}) = \sin\left(\frac{\mathrm{pos}}{10000^{\frac{2\mathrm{i}}{d_{\mathrm{model}}}}}\right) \text{если } i \text{ четное}, \cos\left(\frac{\mathrm{pos}}{10000^{\frac{2\mathrm{i}}{d_{\mathrm{model}}}}}\right) \text{если i нечетное}, \quad (3)$$

где pos – позиция слова в последовательности;

i – индекс измерения вектора;

d_{model} – размерность вектора [38].

АРХИТЕКТУРА ТРАНСФОРМЕРА

На рисунке 1 показана архитектура трансформера, состоящая из двух основных компонентов: кодировщика (encoder) и декодера (decoder). Кодировщик состоит из N одинаковых слоев, каждый из которых содержит механизм многоголового самовнимания, а также промежуточные полносвязные слои и слои нормализации. Декодер также состоит из N одинаковых слоев, но включает дополнительный механизм многоголового внимания, предназначенный для взаимодействия с выходом кодировщика [8].

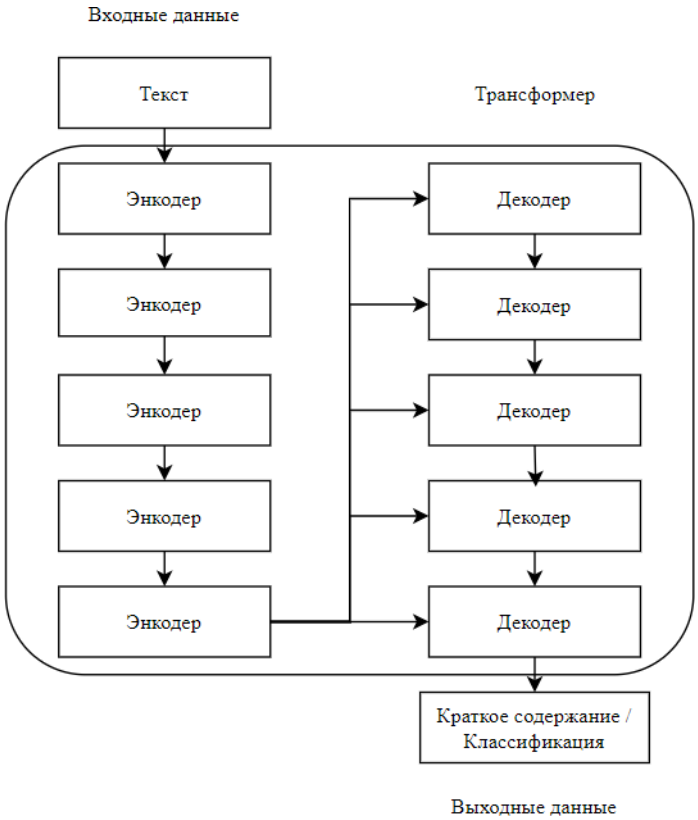


Рисунок 1 – Структура трансформера

Модели, основанные на трансформерах, могут быть предобучены на больших вырезках текста в режиме обучения без учителя (unsupervised learning) или полунадзируемого обучения (semi-supervised learning). Затем они могут быть адаптированы для решения конкретных задач классификации с использованием методов, таких как fine-tuning или zero-shot learning [9].

ПОДГОТОВКА ДАННЫХ

Прежде всего, данные должны быть предобработаны и приведены к формату, подходящему для обучения модели трансформера. Входные данные представляют собой набор текстовых документов, каждый из которых содержит текст и соответствующий класс.

Текстовые данные могут быть предобработаны следующим образом:

– токенизация: текст разбивается на отдельные слова или подслова (токены);

– подготовка словаря: составляется словарь из уникальных токенов, присутствующих в наборе данных;

– конвертация токенов в индексы: токены заменяются соответствующими индексами из словаря;

– паддинг: последовательности дополняются нулями до максимальной длины для создания однородных размеров входных данных [9].

ФАЙН-ТЮНИНГ ТРАНСФОРМЕРА ДЛЯ КЛАССИФИКАЦИИ

Трансформеры, такие как BERT или RoBERTa, предобучены на больших вырезках текста и имеют хорошие знания о языке и контексте. Для адаптации предобученной модели к конкретной задаче классификации используется метод, называемый файн-тюнингом.

При файн-тюнинге модель дообучается на меньшем наборе размеченных данных, содержащих тексты и соответствующие классы. Затем используются выходные данные модели для получения вероятностей принадлежности к каждому классу.

Для адаптации архитектуры трансформера к задаче классификации можно добавить полносвязный слой с размерностью, равной количеству классов, к выходу модели. Затем применяется функция активации softmax (3).

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \text{ для } i = 1, \dots, K \text{ и } z = (z_1, \dots, z_K) \in R^K, \quad (4)$$

где z_i – i-тый элемент входного вектора z;

n – количество элементов в входном векторе;

e – основание натурального логарифма (приближенно равное 2.71828) [9].

Функция softmax гарантирует, что все выходные значения будут находиться в диапазоне от 0 до 1 и их сумма будет равна 1, что делает их интерпретируемыми как вероятности.

После обучения модели необходимо оценить ее производительность на тестовых данных. Для этого можно использовать метрики, такие как точность (accuracy), точность (precision), полноту (recall) и F1-меру (F1-score). Эти метрики позволяют измерить качество работы модели на задаче классификации.

Точность (accuracy) вычисляется как отношение правильно классифицированных текстов к общему числу текстов (4):

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \quad (5)$$

Точность (precision) и полнота (recall) оценивают качество работы модели для каждого класса (5):

$$\text{Precision} = \frac{TP}{TP + FP}, \quad (6)$$

где True Positives (TP) – количество правильно определенных положительных примеров;

False Positives (FP) – количество неправильно определенных положительных примеров;

False Negatives (FN) – количество неправильно определенных отрицательных примеров.

F1-мера представляет собой среднее гармоническое между точностью и полнотой (6):

$$F_1 = 2\frac{(PR)}{(P + R)} \quad (7)$$

где P – Precision;

R – Recall [38].

F1-мера является полезной метрикой, когда классы несбалансированы или когда одновременно важны и точность, и полнота.

Для успешного применения трансформеров к задаче классификации неструктурированной текстовой информации следует учесть следующие аспекты:

– выбор архитектуры трансформера: необходимо определить, какая архитектура трансформера (BERT, GPT, RoBERTa и т. д.) лучше всего подходит для задачи и доступных данных;

– размерность модели и число слоев: более крупные модели и большее количество слоев могут привести к лучшей производительности, но также потребуют больше вычислительных ресурсов и времени на обучение;

– регуляризация: необходимо использовать методы регуляризации, такие как «dropout», для предотвращения переобучения модели;

– размер батча: необходимо определить оптимальный размер батча, чтобы обеспечить эффективное обучение и использование ресурсов [38].

ЭКСПЕРИМЕНТАЛЬНАЯ ЧАСТЬ

Для проведения эксперимента была выбрана задача классификации отзывов о продуктах на основе набора текстов, состоящего из 15 тысяч файлов. Данный набор содержит отзывы пользователей, размеченные на две категории: положительные и отрицательные.

Подготовка данных. Был разделён набор отзывов на обучающую и тестовую выборки в соотношении 80 % на 20 %. Далее проведена предобработка текста. Затем текст был преобразован в числовые последовательности. Также был собран словарь, состоящий из 5000 наиболее часто встречающихся слов. Максимальная длина последовательности была ограничена 128 словами.

ОБУЧЕНИЕ МОДЕЛИ

Была использована предобученная модель BERT и RoBERTa для классификации отзывов. Данные модели различаются в своих подходах к обучению. BERT использует две задачи предварительного обучения: задачу предсказания следующего предложения (Next Sentence Prediction, NSP) и задачу маскированного языкового моделирования (Masked Language Model, MLM). RoBERTa отказывается от задачи NSP, обучаясь только на задаче MLM. Это позволяет модели лучше изучить контекстуальные отношения между словами.

На вершине основной модели BERT был добавлен линейный слой для классификации. Модель была дообучена на обучающей выборке с использованием оптимизатора Adam, скорости обучения 2e-5 и размера батча 32. Всего было проведено 3 эпохи обучения. В процессе обучения была использована кросс-энтропийная функция потерь.

Аналогичено дообучалась и модель RoBERTa. Существенным различием является лишь большее количество эпох – 10.

После обучения моделей была проведена оценка их производительности на тестовой выборке, используя метрики Accuracy, Precision, Recall и F1-мера. Результаты для модели BERT и RoBERTa показаны на рисунке 2.

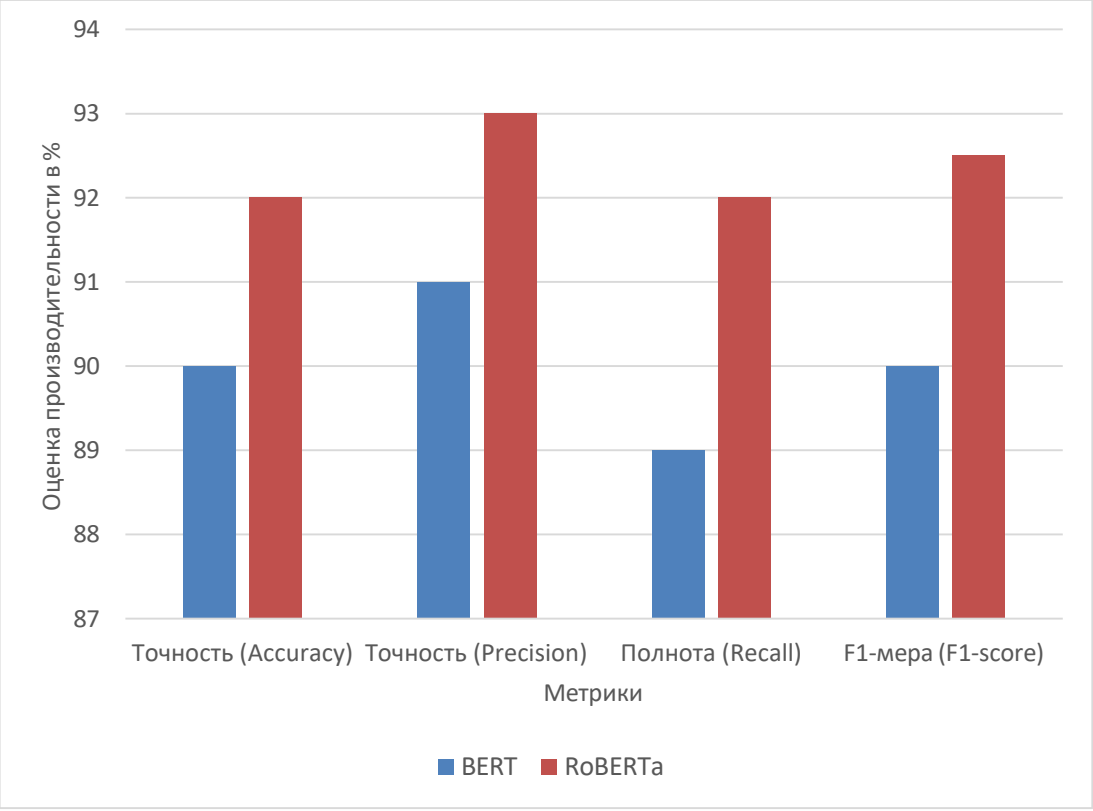


Рисунок 2 – Оценка производительности на основе метрик

Результаты эксперимента показывают, что предобученные модели BERT и RoBERTa способны эффективно классифицировать отзывы о продуктах на основе наборов текстовых отзывов. Для наглядности в таблице представлены показатели метрик моделей BERT, RoBERT, а также сверточных (CNN) и рекурентных (RNN) нейронный сетей.

Таблица – Показатели метрик нейросетевых моделей на тестовой выборке

Модель	Точность (Accuracy)	Точность (Precision)	Полнота (Recall)	F1-score
BERT	90 %	91 %	89 %	90 %
RoBERTa	92 %	93 %	92 %	92,5 %
CNN	85 %	86 %	84 %	85 %
RNN	83 %	85 %	82 %	83,5%

Сверточные и рекуррентные нейронные сети были одними из первых успешно примененных моделей для обработки текстовых данных, и они действительно могут показывать хорошие результаты на многих задачах. Однако модели трансформеров, такие как BERT и RoBERTa, обычно превосходят их благодаря своим уникальным свойствам, таким как возможность эффективно моделировать долгосрочные зависимости в данных и лучшее использование контекстной информации.

Однако стоит отметить, что для улучшения производительности модели можно провести дополнительные эксперименты с различными гиперпараметрами, архитектурами трансформера и методами предобработки данных. Также можно использовать другие предобученные модели для сравнения результатов и определения наилучшей конфигурации для задачи классификации.

ЗАКЛЮЧЕНИЕ

В данной статье было рассмотрено применение моделей трансформера, в частности BERT и RoBERTa, для задачи классификации неструктурированной текстовой информации. Эксперименты показали, что модель BERT способна эффективно классифицировать отзывы о продуктах с высокими значениями метрик производительности.

Возможности и гибкость моделей трансформера открывают перспективы для дальнейших исследований и применения в различных задачах анализа текста. Благодаря использованию предобученных моделей и методам файн-тюнинга, разработчики и исследователи могут значительно сократить время обучения и требуемые вычислительные ресурсы, делая трансформеры доступными для широкого круга пользователей.

Список источников

1. Гудфеллоу, И. Глубокое обучение / И. Гудфеллоу, Й. Бенжио, А. Курвилль. – Москва : Мир, 2018.

2. Китов, В. В. Машинное обучение и анализ данных / В. В. Китов. – Москва : МЦНМО, 2016.

3. Кудрин, Н. Д. Нейронные сети и глубокое обучение / Н. Д. Кудрин. – Москва : ФИЗМАТЛИТ, 2017.

4. Лопатин, А. В. Технологии обработки и анализа текстовых данных с использованием глубокого обучения / А. В. Лопатин // Управление большими системами. – 2017. – № 66. – С. 136–152.

5. Бухановский, А. В., Старостин А. М., Царёв А. В. Анализ текстовых данных с помощью нейросетевых алгоритмов / А. В. Бухановский, А. М. Старостин, А. В. Царёв // Известия ЮФУ. Технические науки. – 2016. – № 9 (180). – С. 15–27.

6. Школа глубокого обучения. Физтех. URL: https://www.dlschool.org/ (дата обращения: 12.05.2023).

7. Специализация «Машинное обучение и анализ данных» // Coursera. – URL: https://www.coursera.org /specializations/machine-learning-data-analysis (дата обращения: 12.05.2023).

8. Николенко, С. Нейронные сети и глубокое обучение / С. Николенко, А. Кадурин, Е. Архангельская. – Санкт-Петербург : Питер, 2018.

9. Барсегян, А. А. Технологии анализа текста / А. А. Барсегян, С. В. Купцов, А. А. Лошкарев // Системы и средства информатики. – 2011. – № 21. – С. 170–198.

10. Специализация «Нейронные сети для обработки текстов» // Coursera. – URL: https://www.coursera.org/specializations/neural-networks-for-text-processing (дата обращения: 12.05.2023).

11. Vaswani, A. Attention is all you need / A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez & I. Polosukhin // Advances in neural information processing systems. – 2017. – P. 5998–6008.

12. Devlin, J. BERT: Pre-training of deep bidirectional transformers for language understanding / J. Devlin, M. W. Chang, K. Lee & K. Toutanova // arXiv preprint arXiv. – 2018. – 1810.04805.

13. Radford, A. Improving language understanding by generative pre-training / A. Radford, K. Narasimhan, T. Salimans & I. Sutskever. – 2018. – URL https://s3-us-west-2.amazonaws.com/openai-assets/researchcovers/languageunsupervised/languageunderstandingpaper.pdf.

14. Liu, Y. RoBERTa: A robustly optimized BERT pretraining approach / Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen & V. Stoyanov // arXiv preprint arXiv. – 2019. – 1907.11692.

15. Wolf, T. Huggingface's transformers: State-of-the-art natural language processing / T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi & A. M. Rush // arXiv preprint arXiv. – 2020. – 1910.03771.

16. Brown, T. B. Language models are few-shot learners / T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal & S. Agarwal // arXiv preprint arXiv. – 2020. – 2005.14165.

17. Pennington, J. Glove: Global vectors for word representation / J. Pennington, R. Socher & C. Manning // Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). – 2014. – P. 1532–1543.

18. Mikolov, T. Efficient estimation of word representations in vector space / T. Mikolov, K. Chen, G. Corrado & J. Dean // arXiv preprint arXiv. – 2013. – 1301.3781.

19. Lample, G. Cross-lingual language model pretraining / G. Lample & A. Conneau // arXiv preprint arXiv. – 2019. – 1901.07291.

20. Scikit-learn: Machine learning in Python, Pedregosa et al. // JMLR 12. – 2011.– P. 2825–2830.

References

1. Goodfellow, I., Bengio, Y., Courville, A. *Deep learning.* Moscow, Mir, 2018 (In Russ.).

2. Kitov, V. V. *Machine learning and data analysis.* Moscow, MTsNMO, 2016 (In Russ.).

3. Kudrin, N. D. *Neural networks and deep learning.* Moscow, FIZMATLIT, 2017 (In Russ.).

4. Lopatin, A. V. Technologies for processing and analyzing text data using deep learning. *Management of large systems*, 2017, no. 66, pp. 136–152 (In Russ.).

5. Bukhanovsky, A. V., Starostin, A. M., Tsarev, A. V. Analysis of text data using neural network algorithms. *News of the Southern Federal University. Technical science*, 2016, no. 9 (180), pp. 15–27 (In Russ.).

6. *School of deep learning. Phystech.* URL: https://www.dlschool.org/ (accessed 05.12.2023) (In Russ.).

7. Specialization “Machine learning and data analysis”. *Coursera.* URL: https://www.coursera.org/specializations/machine-learning-data-analysis (accessed 05.12.2023) (In Russ.).

8. Nikolenko, S., Kadurin, A., Arkhangelskaya, E. *Neural networks and deep learning.* St. Petersburg, Peter, 2018 (In Russ.).

9. Barseghyan, A. A., Kuptsov, S. V., Loshkarev, A. A. Technologies for text analysis. *Systems and means of informatics*, 2011, no. 21, pp. 170–198 (In Russ.).

10. Specialization “Neural networks for text processing”. *Coursera*. URL: https://www.coursera.org/specializations/neural-networks-for-text-processing (accessed 05.12.2023). (In Russ.).

11. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N. & Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 2017, pp. 5998–6008.

12. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv*, 2018, 1810.04805.

13. Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. Improving language understanding by generative pre-training. 2018. URL https://s3-us-west-2.amazonaws.com/openai-assets/researchcovers/languageunsupervised /languageunderstandingpaper.pdf.

14. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D. & Stoyanov, V. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv*, 2019, 1907.11692.

15. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., & Rush, A. M. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv*, 2020, 1910.03771.

16. Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P. & Agarwal, S. Language models are few-shot learners. *arXiv preprint arXiv*, 2020, 2005.14165.

17. Pennington, J., Socher, R., & Manning, C. Glove: Global vectors for word representation. *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

18. Mikolov, T., Chen, K., Corrado, G., & Dean, J. Efficient estimation of word representations in vector space. *arXiv preprint arXiv*, 2013, 1301.3781.

19. Lample, G., & Conneau, A. Cross-lingual language model pretraining. *arXiv preprint arXiv*, 2019, 1901.07291.

20. Scikit-learn: Machine learning in Python, Pedregosa et al. *JMLR 12*, 2011, pp. 2825–2830.

Статья поступила в редакцию 08.04.2024; одобрена после рецензирования 26.04.2024; принята к публикации 14.05.2024.

The article was submitted 08.04.2024; approved after reviewing 26.04.2024; accepted for publication 14.05.2024.