Моделирование и анализ данных
2024. Том 14. № 1. С. 7–26
DOI: https://doi.org/10.17759/mda.2024140101
ISSN: 2219-3758 (печатный)
ISSN: 2311-9454 (online)

Modelling and Data Analysis
2024. Vol. 14, no. 1, pp. 7–26
DOI: https://doi.org/10.17759/mda.2024140101
ISSN: 2219-3758 (print)
ISSN: 2311-9454 (online)

◇◇◇◇◇◇◇◇◇◇АНАЛИЗ ДАННЫХ◇◇◇◇◇◇◇◇◇◇

УДК 004.85

Семантический анализ отзывов об организациях методами машинного обучения

Платонов Е.Н.*
Московский авиационный институт
(национальный исследовательский университет)
(ФГБОУ ВО МАИ), г. Москва, Российская Федерация
ORCID: https://orcid.org/0000-0001-8502-1350
e-mail: en.platonov@gmail.com

Мартынова И.Р.**
Московский авиационный институт
(национальный исследовательский университет)
(ФГБОУ ВО МАИ), г. Москва, Российская Федерация
ORCID: https://orcid.org/0009-0007-3140-2490
e-mail: irina.mart.r@gmail.com

Семантический анализ отзывов организаций представляет собой ключевой инструмент для оценки уровня удовлетворенности клиентов. Бизнес-структуры должны регулярно проводить анализ и исследование эмоционального фона с целью углубленного изучения данных и более полного понимания своей деятельности, в том числе с помощью методов машинного обучения. Так же настоящее время, методы, основанные на глубоком обучении, привлекают повышенное внимание благодаря их высокой эффективности. В данном исследовании мы сосредоточимся на задаче определения тональности текста. Для выполнения сентимент-анализа мы прибегнем к методам машинного обучения, включая различные подходы к векторному преобразованию текста, модели глубокого обучения и алгоритмы обработки естественного языка (NLP).

Ключевые слова: обработка текстов на естественном языке, задача классификации, градиентный бустинг, рекуррентные нейронные сети, сверточные нейронные сети, Bert, GPT.

Для цитаты:
Платонов Е.Н., Мартынова И.Р. Семантический анализ отзывов об организациях методами машинного обучения // Моделирование и анализ данных. 2024. Том 14. № 1. С. 7–26. DOI: https://doi.org/10.17759/mda.2024140101

***Платонов Евгений Николаевич,** кандидат физико-математических наук, доцент, Московский авиационный институт (национальный исследовательский университет) (ФГБОУ ВО МАИ), г. Москва, Российская Федерация, ORCID: https://orcid.org/0000-0001-8502-1350, e-mail: en.platonov@gmail.com

****Мартынова Ирина Романовна,** студент магистратуры института «Информационные технологии и прикладная математика» Московский авиационный институт (национальный исследовательский университет), г. Москва, Российская Федерация, ORCID: https://orcid.org/0009-0007-3140-2490, e-mail: irina.mart.r@gmail.com

1. ВВЕДЕНИЕ

Обработка естественного языка (NLP) является областью, которая использует вычислительные методы для анализа и синтеза естественного языка и речи. Эта дисциплина пересекается с лингвистикой, изучая слова и языковые конструкции, и тесно взаимодействует с компьютерными науками и искусственным интеллектом.

NLP не только исследует основы языка, но и успешно применяется для разнообразных задач, начиная от поддержки образовательного процесса [1, 2] и заканчивая исследованиями в области психиатрии, таких как изучение пациентов с шизофренией [3]. Интересно, что NLP также находит применение в неожиданных областях, включая анализ рентгенограмм легочной ткани [4].

Благодаря обработке естественного языка открываются уникальные возможности в решении различных задач, включая анализ настроений [5]. Один из подходов, предложенный в работе [6], использует глубокое обучение для анализа сообщений в социальных сетях с помощью векторных представлений слов (например, fastText, word2vec и GloVe) и сверточных нейронных сетей (CNN).

Другие исследователи [7] разработали новую систему рекомендаций продуктов, основанную на инновационной логике, которая предсказывает наиболее значимые продукты для онлайн-покупок в реальном времени, учитывая предпочтения клиентов и оценки продуктов. Еще одно исследование [8] показало, что, разделяя отзывы на 8 эмоциональных категорий, можно проанализировать поведение клиентов и улучшить их уровень удовлетворенности. Также проводились исследования с применением сложных архитектур нейронных сетей, таких как BERT [9, 10].

В данной работе мы стремимся заполнить пробел в исследованиях семантического анализа текста на русском языке с использованием современных нейросетевых архитектур. Мы сосредотачиваемся на анализе тональности отзывов об организациях на Яндекс Картах. Наша цель – разработать модель, автоматически классифицирующую отзывы как положительные, отрицательные или нейтральные, и оценить ее эффективность на реальных данных. Мы подготавливаем корпус текстов отзывов, включающий положительные, нейтральные и отрицательные отзывы, затем модель обучается на корпусе с использованием различных алгоритмов, и в конце происходит сравнение результатов.

2. ВЕКТОРНОЕ ПРЕДСТАВЛЕНИЕ ТЕКСТА

Теоретической базой для векторного представления слов считается дистрибутивная семантика. Она изучает семантическую близость слов на основе их распределения в текстовых корпусах [11]. Это позволяет перевести информацию о словах в многомерные векторы с помощью линейной алгебры, что обеспечивает их компактное представление для вычислений [12].

TF-IDF

TermFrequency-Inverse Document Frequency – это метод представления слов в векторной форме, учитывающий не только частоту встречаемости слова в документе (TF), но и его важность в контексте всего корпуса (IDF) [13].

TF (Частота термина): отражает, насколько часто слово встречается в документе. Вычисляется как отношение числа вхождений слова к общему числу слов в документе.

$$tf\left(t,d\right)=\frac{n_t}{\sum_k n_k},$$

где n_t – число вхождений слова в документ, $\sum_k n_k$ – общее число слов в данном документе.

IDF (Обратная частота документа): показывает, насколько уникально слово в контексте всего корпуса. Вычисляется как логарифм обратного отношения общего числа документов к числу документов, содержащих слово.

$$idf\left(t,D\right)=\log\frac{\left|D\right|}{|\{d_i\in D\,|\,t\in d_i\}|},$$

где |D| – число документов в коллекции;
$\left|\{d_i\in D|t\in d_i\}\right|$ – число документов из коллекции D, в которых встречаться t.

Word2vec

Модель word2vec базируется на предположении о тесной связи между словами и частями речи в их векторных представлениях [14]. Её архитектура, основанная на контекстной близости слов, обычно включает следующие этапы: 1) токенизация словаря; 2) представление корпуса слов в векторной форме через one-hotencoding; 3) ввод полученных векторов слов в нейронную сеть; 4) расчет ошибки через вычисление функции потерь; 5) коррекция весов модели с использованием обратного распространения ошибки.

Word2vec предлагает две различные архитектуры: Skip-gram (по слову предсказывается контекст) и CBOW (по контексту предсказывается слово) [15].

Для модели Skip-gram условная вероятность генерации любого центрального слова с учетом окружающих слов контекста может быть смоделирована по следующе формуле:

$$P(w_c\mid w_o)=\frac{exp\left(u_c^T\overline{v_0}\right)}{\sum i\in v\,exp\left(u_i^T\overline{v_0}\right)},$$

где w_c – центральное слово, w_o – контекст, v_i и u_i – вектора центрального слова и контекста соответственно, v – размер словаря.

Параметрами модели skip-gram являются вектор центрального слова и вектор контекстного слова для каждого слова в словаре. В процессе обучения мы изучаем эти параметры, максимизируя функцию правдоподобия, что эквивалентно минимизации функции потерь:

$$\mathcal{L} = \sum_{t=1}^{T} \sum_{-c \le j \le c, j \ne 0} \log p\left(w_{t+j} | w_t\right),$$

где T – это общее количество слов в корпусе, c – это размер окна, w_t - центральное слово в позиции t, w_{t+j} - – контекстное слово в позиции $t+j$, а $p(w_{t+j} \mid w_t)$ - это вероятность того, что слово w_{t+j} будет встречаться в контексте слова w_t.

Часто эта функция правдоподобия реализуется с помощью бинарного кросс--энтропийного лосса:

$$\mathcal{L} = \sum_{t=1}^{T} \sum_{-c \le j \le c, j \ne 0} y_{t,j} \log p_{t,j} + \left(1 - y_{t,j}\right) \log\left(1 - p_{t,j}\right),$$

где $y_{t,j}$ – это целевая вероятность того, что слово w_{t+j} будет встречаться в контексте слова w.

CBOW – метод обучения модели на основе текста, предсказывающий слово по его контексту. Формализуется следующим образом:

$$P(w_o \mid w_c) = \frac{exp\left(u_c^T \overline{v_0}\right)}{\sum i \in v exp\left(u_i^T \overline{v_0}\right)},$$

где w_c – центральное слово, w_o – контекст, v_i и u_i – вектора центрального слова и контекста соответственно, v – размер словаря.

Обучение CBOW аналогично обучению skip-gram. Оценка максимального правдоподобия модели CBOW эквивалентна минимизации функции потерь:

$$\mathcal{L} = -\sum_{i=1}^{V} y_i \, log \, p(w_i \mid w_I) = -\log p\left(w_o | w_I\right),$$

где $(w_o | w_I)$ – это целевая вероятность.

Классическая модель word2vec имеет проблемы с редкими словами, особенно в морфологически богатых языках, таких как французский, испанский и русский.

3. МЕТОДЫ

В работе применяются методы машинного обучения, такие как логистическая регрессия, градиентный бустинг (CatBoost), метод опорных векторов (SVM), мультиномиальный наивный байес, а также методы глубокого обучения, включая рекуррентные

нейронные сети долговременной краткосрочной памяти (LSTM), сверточные нейронные сети (CNN) и трансформеры (Bert и GPT).

Логистическая регрессия

Логистическая регрессия применяется для прогнозирования вероятности принадлежности объекта к определенному классу. В многоклассовой классификации используется метод «Один против остальных» (One-vs-Rest, OvR), где создается K бинарных классификаторов (где K – количество классов), каждый из которых отличает один класс от всех остальных:

$$P\left(Y=k\right)=\frac{e^{\left(\beta_{0k}+\beta_{1k}X_1+\beta_{2k}X_2+\ldots+\beta_{nk}X_n\right)}}{\sum_{j=1}^{K}e^{\left(\beta_{0j}+\beta_{1j}X_1+\beta_{2j}X_2+\ldots+\beta_{nj}X_n\right)}},$$

где P(Y=k) – вероятность принадлежности к классу k,
K – общее количество классов,
β_{0k}, β_{1k}, …, β_{nk} – коэффициенты регрессии для класса k,

Модель выбирает класс с наибольшей предсказанной вероятностью.

Метод опорных векторов

Метод опорных векторов или SVM (Support Vector Machines) – линейный алгоритм для классификации и регрессии. Применяется широко и для линейных, и для нелинейных задач. SVM строит классифицирующую функцию F в виде:

$$F\left(x\right)=sign(\left\{w,x\right\}+b,$$

где $\{\cdot,\cdot\}$ – скалярное произведение, $\boldsymbol{w}$ – нормальный вектор к разделяющей гиперплоскости, b – свободный член.

Мы стремимся выбрать w и b так, чтобы максимизировать расстояние до каждого класса. Можно подсчитать, что данное расстояние равно $\frac{1}{|w|}$. Проблема нахождения максимума $\frac{1}{|w|}$ эквивалентна проблеме нахождения минимума $|w|^2$. Запишем все это в виде задачи оптимизации:

$$\begin{cases} arg\,min\left|w\right|^2 \\ y_i\left(w,x_i+b\right)\geq 1, i=1,\ldots,m. \end{cases}$$

Оптимизационная задача сводится к минимизации суммы квадратов весов для каждого бинарного классификатора с соответствующими ограничениями.

Градиентный Бустинг (CatBoostClassifier)

CatBoostClassifier – алгоритм машинного обучения на основе градиентного бустинга деревьев решений. Эффективно справляется с задачами классификации и регрессии [16], снижает переобучение и использует весь набор данных для обучения.

Реализует случайную перестановку данных и вычисляет среднее значение меток для каждого элемента в соответствии с предшествующими элементами в перестановке. Предположим, что у нас имеется набор наблюдений $D = \{(X_i, Y_i)\}_{i=1\ldots n}$, где $X_i = (x_{i,1}, \ldots, x_{i,m})$ представляет собой вектор из m признаков, включающих как числовые, так и категориальные, а $Y_i \in R$ – это целевая переменная. Обозначим перестановку как $\sigma = (\sigma_1, \ldots, \sigma_n)$. Тогда значение $x_{\text{ср},k}$ заменяется соответствующим средним значением.

$$\frac{\sum_{j=1}^{p-1}\left[x_{\text{ój},k} = \boldsymbol{x}_{\text{óp},k}\right]\boldsymbol{Y}_{\text{ój}} + \boldsymbol{aP}}{\sum_{j=1}^{p-1}\left[x_{\text{ój},k} = \boldsymbol{x}_{\text{óp},k}\right] + \boldsymbol{a}},$$

здесь вводится предварительное значение P и параметр $a > 0$, который представляет собой вес предварительного значения. Добавление предварительного значения – распространенная практика, способствующая снижению шума, возникающего из-за категорий с низкой частотой.

Мультиномиальный наивный Байес

В мультиномиальной модели документ рассматривается как последовательность случайных выборов слов из «мешка слов». Правдоподобие документа оценивается через вероятности совпадения выбранных слов с теми, что встречаются в документе. Наивное предположение заключается в том, что выбор каждого слова из мешка происходит независимо от выбора других слов.

В математической модели, где $V = \{\text{wt}\}_{t=1}^{|V|}$ представляет собой словарь, каждый документ d_i представляется в виде вектора длины $|d_i|$, состоящего из слов, каждое из которых «вынуто» из словаря с вероятностью $p(w_t|c_j)$. Правдоподобие принадлежности документа d_i классу c_j определяется следующим образом:

$$p\left(d_i|c_j\right) = p\left(\left|d_i\right|\right)\left|d_i\right|!\prod_{t=1}^{|V_i|}\frac{1}{N_{it}!}p\left(w_t|c_j\right)^{N_{it}},$$

где N_{it} – количество вхождений w_t в d_i.

Для обучения такого классификатора тоже нужно обучить вероятности $p(w_t \mid c_j)$. Оптимальные оценки вероятностей встречаемости слова w_t в классе c_j могут быть вычислены, учитывая набор документов $D = \{d_i\}_{i=1}^{|D|}$, уже распределенных по классам c_j, возможно, даже вероятностно. Дан словарь $V = \text{wt}_{t=1}^{|V|}$, и известны вхождения N_{it}. Оптимальные оценки могут быть выражены с учетом сглаживания Лапласа:

$$p\left(w_t|c_j\right) = \frac{1 + \sum_{i=1}^{|D|}N_{it}\,p\left(c_j|d_i\right)}{|V| + \sum_{s=1}^{|V|}\sum_{i=1}^{|D|}N_{is}\,p\left(c_j|d_i\right)},$$

Априорные вероятности классов можно посчитать как:

$$p\left(c_j\right) = \frac{1}{|D|}\sum_{i=1}^{|D|}p\left(c_j|d_i\right),$$

Тогда классификация будет проходить как:

$$c = \arg\max_j \left[\log\left(\frac{1}{|D|} \sum_{i=1}^{|D|} p\left(c_j | d_i\right) \right) + \sum_{t=1}^{|V|} N_{it} \log p\left(w_t | c_j\right) \right]$$

Сверточные нейронные сети (CNN)

Сверточные нейронные сети (CNN) – класс сетей, разработанных для обработки структурированных данных, в первую очередь изображений, широко применяются в компьютерном зрении [17]. Они также могут использоваться для обработки текстовых данных, извлекая признаки из последовательности слов или символов.

В текстовых задачах CNN текст представляется матрицей встраивания, где каждое слово или символ представлено вектором. Сверточные слои применяются для извлечения локальных признаков, используя фильтры разных размеров окна. Затем выполняется операция объединения (pooling), например, максимальное или среднее объединение, для уменьшения размерности признаков. Полученные признаки передаются в полносвязные слои для классификации или других задач обработки текста.

CNN выделяют локальные паттерны и последовательности слов, что помогает модели анализировать различные языковые структуры, такие как n-граммы и контексты. Это полезно для классификации текста, анализа тональности, определения языка и других задач обработки.

Рекуррентные нейронные сети (RNN)

Рекуррентные нейронные сети (RNN) – модели глубокого обучения, которые обрабатывают последовательности с использованием рекуррентных связей, передающих информацию между временными шагами. Они развертываются по времени, применяя одни и те же параметры на каждом шаге, в отличие от стандартных сетей, которые передают информацию синхронно на каждом слое.

Сеть долговременной краткосрочной памяти LSTM (Long Short-Term Memory) была создана чтобы нивелировать недостатки базовой рекуррентной нейронной сети, такие как взрыв и затухание градиентов и проблемы с памятью сети [18].

Рассмотрим ячейку памяти LSTM:

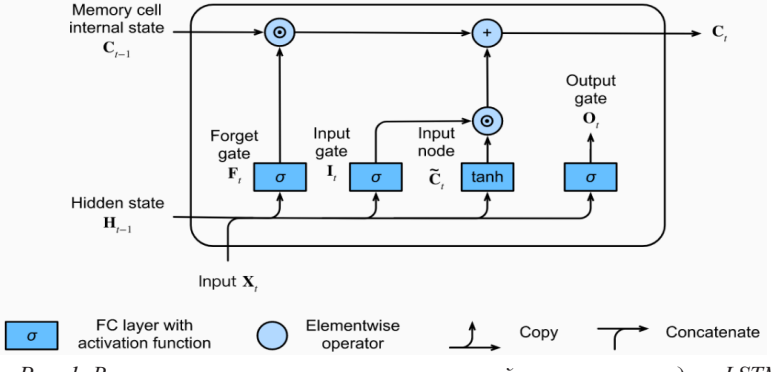

Рис. 1. Вычисление внутреннего состояния ячейки памяти в модели LSTM

LSTM включает в себя три вида шлюзов (Gate): входной, забывания и выходной. Выход скрытого слоя LSTM состоит из скрытого состояния и внутреннего состояния ячейки памяти. Только скрытое состояние передается на выходной слой, в то время как внутреннее состояние ячейки памяти остается полностью внутренним.

Трансформеры

Трансформер – модель, использующая только механизмы внимания без сверточных или рекуррентных слоев. Популярна в различных областях глубокого обучения, включая языковые задачи, обработку изображений и обучение с подкреплением [19]. Рассмотрим архитектуру:

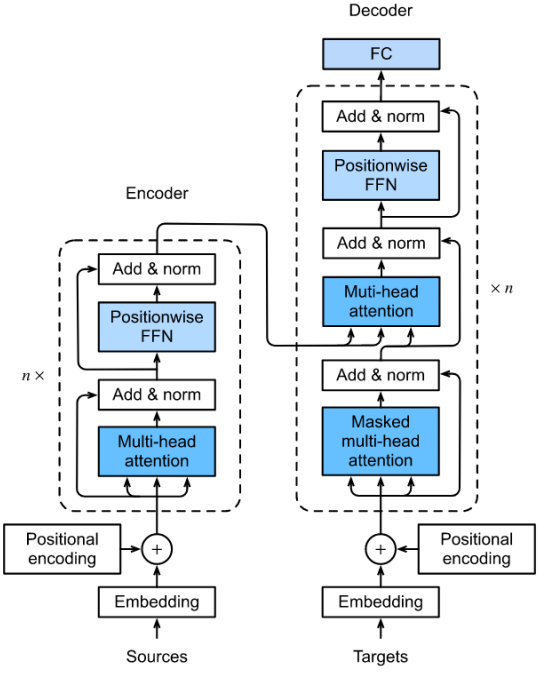

Рис 2. Архитектура трансформера

И одним из важных механизмов работы трансформеров является механизм Self-Attention, его стоит разобрать более подробно. Механизм работы self-attention состоит из того, что на вход подаётся вектор слов x_1, $x_2..x_{n,}$ а на выходе мы получаем вектор представлений z_1, $z_2…z_n$.

Для этого для каждого слова обучают три вектора k = key (ключ, куда мы смотрим), v = value (значение, что мы видим), q = query (запрос, откуда мы смотрим). Таким образом, для каждого слова будет считаться три вектора с помощью вектора весов: $q_j = W_q x_j$, $k_{j=} W_v x_j$, $v_j = W_v x_{j.}$ Важность x_i слова для обновления x_j слова мы

будем считать благодаря скалярному произведению q_j на k_i. Далее мы рассчитываем веса по формуле:

$$w_{ij} = \frac{exp\frac{\langle q_j, k_i \rangle}{\sqrt{d}}}{\sum_{p=1}^{n} exp\frac{\langle q_j, k_p \rangle}{\sqrt{d}}},$$

где d – размерность векторов, n – число слов входной последовательности.

После этого новое представление x_j мы считаем, как:

$$z_j = \sum_{p=1}^{n} w_{pj} v_p .$$

Модификация self-attention – это Multi-head attention (MHA) (рис. 3), где для каждого слова мы пересчитываем его представление несколько раз с разными весами. MHA позволяет модели совместно воспринимать информацию из разных представлений подпространства в разных положениях

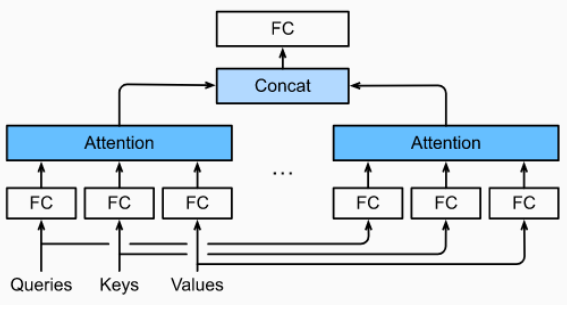

Рис. 3. MHA, где несколько голов объединены, затем линейно преобразованы.

Данная архитектура лежит в основе популярных сетей, таких как Bert (состоящий только из блоков энкодера) и GPT (состоящий только из блоков декодера), которые были использованы в нашем исследовании.

4. РЕЗУЛЬТАТЫ И ОБСУЖДЕНИЯ

Рассматриваем «Geo Reviews Dataset 2023» от Яндекса – более 500 000 отзывов организаций в России, собранных за 2023 год. Ценный ресурс для исследований в области анализа тональности и оценки организаций.

Основные атрибуты включают:

- Адрес организации (address)
- Название организации (name_ru)
- Список рубрик (rubrics)

- Оценка пользователя от 0 до 5 (rating)
- Текст отзыва (text)

При анализе данных было решено переопределить метки классов для достижения более сбалансированного состояния. Исходные пять категорий оценок (от 1 до 5) были объединены с целью упрощения задачи анализа тональности отзывов и обучения моделей.

- *Негативные отзывы:*
  - Оценка 1: сильно негативный отзыв;
  - Оценка 2: негативный отзыв.
- *Нейтральные отзывы:*
  - Оценка 3: нейтральный отзыв.
- *Позитивные отзывы:*
  - Оценка 4: очень позитивный отзыв;
  - Оценка 5: сильно позитивный отзыв.

Приведем часть данных в виде таблицы 1.

Таблица 1

Данные из набора Geo Reviews Dataset

	address	**name_ru**	**rating**	**rubrics**	**text**
0	Московская область, Электросталь, проспект Ленина, 29	Продукты Ермолино	5.0	Магазин продуктов; Продукты глубокой заморозки;	Замечательная сеть магазинов в общем, хороший ассортимент, цены приемлемые, а главное качество на высоте!!! …
…	…	…	…	…	…
…	…	…	…	…	…
500 112	Краснодар, Прикубанский внутригородской округ, микрорайон имени Петра Метальникова, улица Петра Метальникова, 26	LimeFit	1.0	Фитнес-клуб	Не знаю смутят ли кого-то данные правила, но я была удивлена: хочешь, чтобы твой шкаф замыкался – купи замочек.

Слияние оценок в три категории обеспечило более гармоничное распределение между классами и позволило уменьшить несбалансированных классов в данных. В итоговом датасете представлено 117779 строк данных.

Качество решения задачи классификации оценивалось с помощью F1 меры. Формула для взвешенной F1-меры $\left(F1_{\text{weighted}}\right)$ в многоклассовой классификации имеет вид:

$$F1_{\text{weighted}} = \frac{\sum_i w_i F1_i}{\sum_i w_i},$$

где:

- $F1_{\text{weighted}}$ – взвешенная F1-мера,
- $F1_i$ – F1-мера для класса i,
- w_i – вес класса i, рассчитанный как доля объектов класса i от общего числа объектов.

Качество решения задачи так же оценивалось с помощью матрицы ошибок.

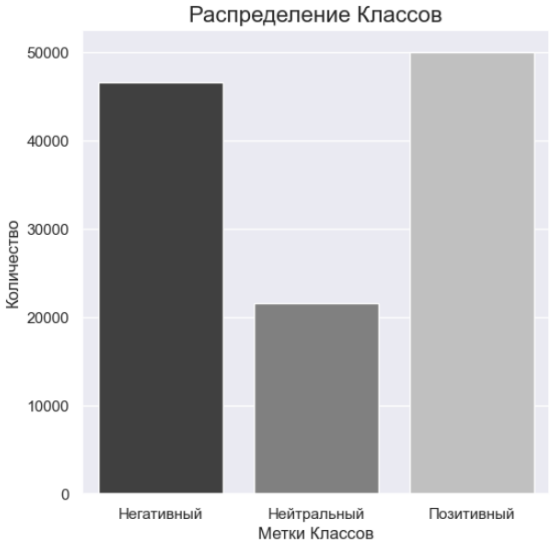

Рис. 4. Распределение меток классов

Таблица 2

Матрица ошибок

Истинный класс		a=0	a=1	a=2
	y=0	Y_{00}	Y_{01}	Y_{02}
	y=1	Y_{10}	Y_{11}	Y_{12}
	y=2	Y_{21}	Y_{22}	Y_{23}
	Предсказанный класс			

В таблице 2 первый индекс означает истинный класс, а второй предсказанный, таким образом правильно предсказанные классы располагаются по диагонали матрицы.

Применение классических методов машинного обучения

При решении поставленной задачи обратимся сначала к фундаментальным алгоритмам классификации используем логистическую регрессию, CatBoost, SVM и мультиномиальный наивный Байес для классификации текстов. Применяем метод TF-IDF для векторизации данных.

Логистическая регрессия

В ходе нашего исследования использовался GridSearchCV из sklearn.model_ selection для подбора параметров. Наилучшие результаты достигнуты с параметрами: lr_C = 0.1, multi_class=’ovr’, max_iter=500, class_weight=’balanced’. Это привело к F1-мере 0.78 и следующей матрице ошибок:

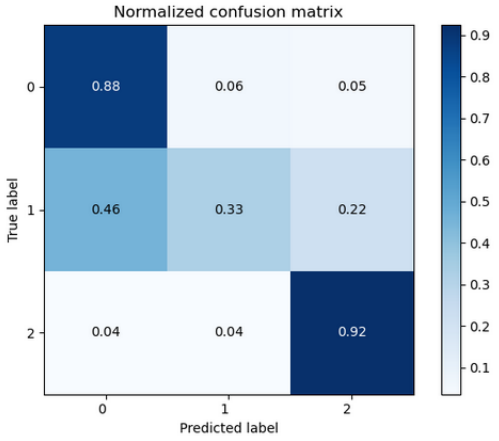

Рис. 5. Матрица ошибок для модели логистической регрессии

Метод опорных векторов

В модели мы использовали метод TF-IDF и дополнительные признаки. С помощью GridSearchCV подобрали оптимальные гиперпараметры: penalty = l2, class_weight = balanced, max_iter = 1000, multi_class='ovr'. Это привело к F1-мере 0.77 и следующей матрице ошибок:

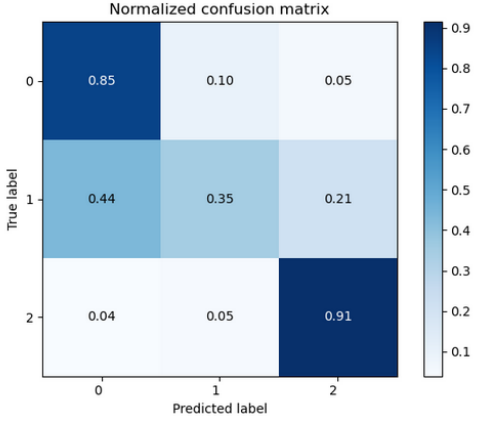

Рис 6. матрица ошибок для модели SVM

Градиентный Бустинг (CatBoostClassifier)

Были использованы следующие гиперпараметры модели: learning_rate = 0.1, depth = 10, iterations = 1000, и l2_leaf_reg = 5. Это привело к F1-мере 0.76 и следующей матрице ошибок:

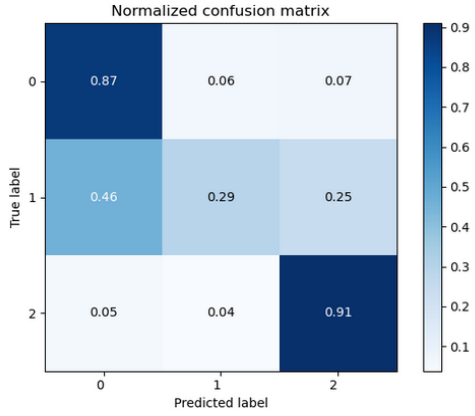

Рис. 7. Матрица ошибок для модели CatBoostClassifier

Мультиномиальный наивный Байес

Модель мультиномиального наивного Байеса была запущена с исходными параметрами: alpha = 0.1, fit_prior = True. В результате этих настроек удалось достичь F1-меры в 0.75, а сформирована матрица ошибок в следующем виде:

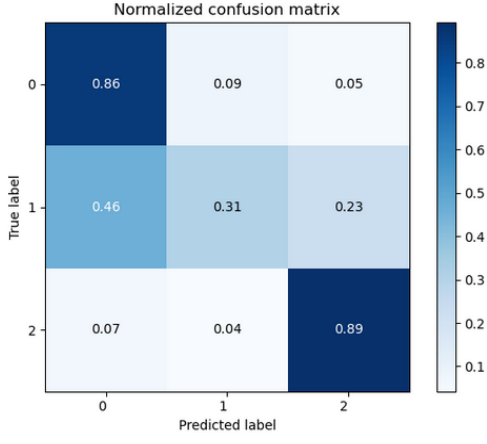

Рис. 8. Матрица ошибок модели мультиномиального наивного Байеса

Применение методов глубокого обучения

Сверточные нейронные сети(CNN)

CNN обучалась с помощью оптимизатора AdamW, lr = 1e-4, batch_size = 512. Использовались Word2Vec эмбеддинги. Это привело к точности с F1-мерой 0.75 и соответствующей матрице ошибок:

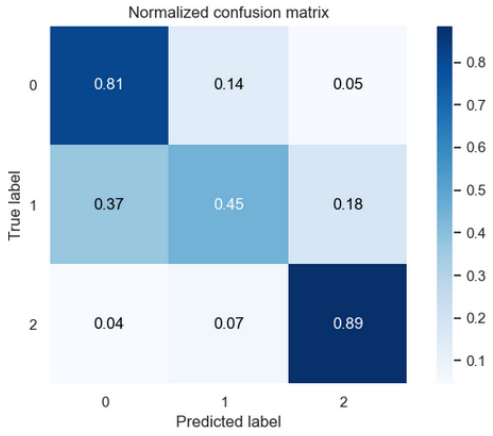

Рис. 9 Матрица ошибок модели CNN

Рекуррентная нейронная сеть (LSTM)

LSTM была оптимизирована с AdamW, lr = 1e-3. Достигнута точность F1-меры 0.79, сформирована матрица ошибок для подробного обзора результатов модели:

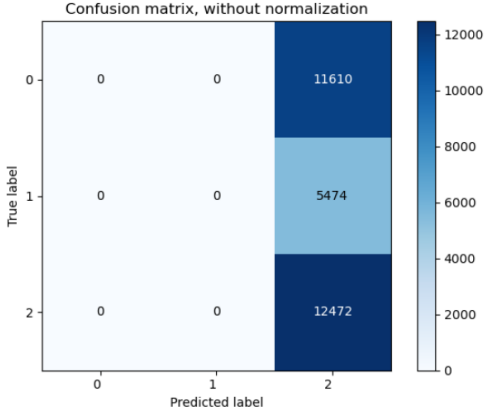

Рис. 10. Матрица ошибок LSTM

Трансформеры

Использовали три варианта модели BERT: базовая версия от DeepPavlov с 12 слоями, rubert-tiny2 с тремя слоями и была предпринята попытка сократить архитектуру до одного слоя и провести обучение на уменьшенной модели. Все архитектуры обучались с применением оптимизатора AdamW, lr = 5e-5. Базовая BERT показала переобучение с низкой F1-мерой 0.26, указывая на необходимость корректировок для более устойчивых результатов:

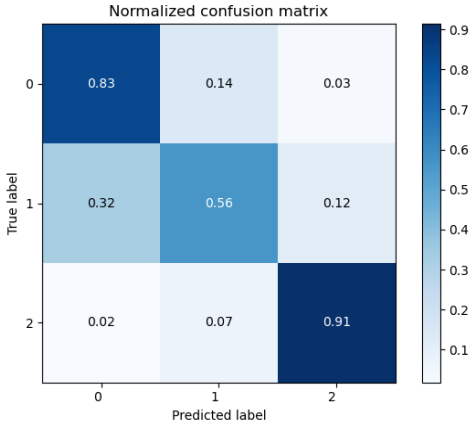

Рис. 11. Матрица ошибок BERT

Модель rubert-tiny2 продемонстрировала выдающуюся производительность, достигнув значения F1-меры на уровне 0.83. В результате этого успешного обучения была сформирована матрица ошибок:

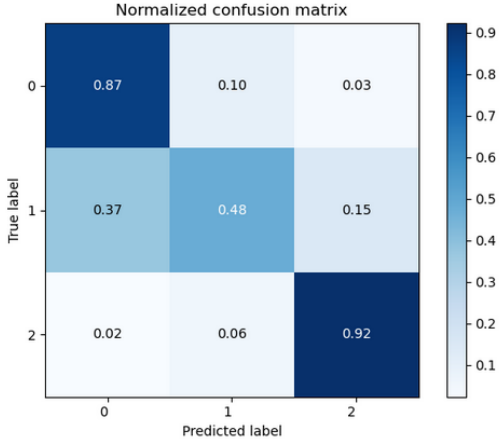

Рис 12. Матрица ошибок rubert-tiny2 (не измененный)

Модель rubert-tiny2 с одним слоем проявила впечатляющую производительность с F1-мерой, достигшей значения 0.81. Этот результат подчеркивает эффективность уменьшенной архитектуры в сравнении с более глубокими моделями (*см. рис. 13*).

Для дополнительного анализа использовалась модель ruGPT3small от Сбербанка с двумя слоями и оптимизатором SophiaG [20], lr = 8e-6. Достигнута F1-мера 0.79, сформирована матрица ошибок:

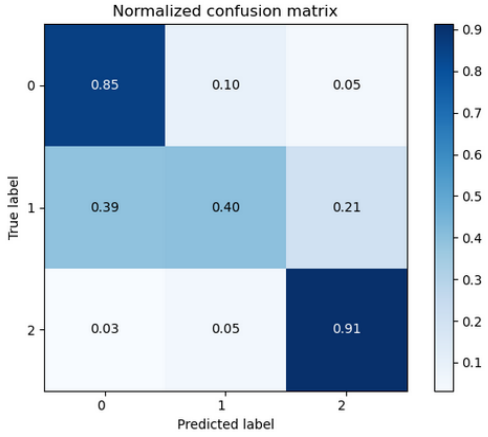

Рис. 14. Матрица ошибок rugpt3small с двумя слоями

Применим алгоритм LIME для интерпретации результатов модели rubert-tiny2 с наивысшей точностью. Этот шаг поможет понять влияние факторов на решения модели, повышая интерпретируемость результатов.

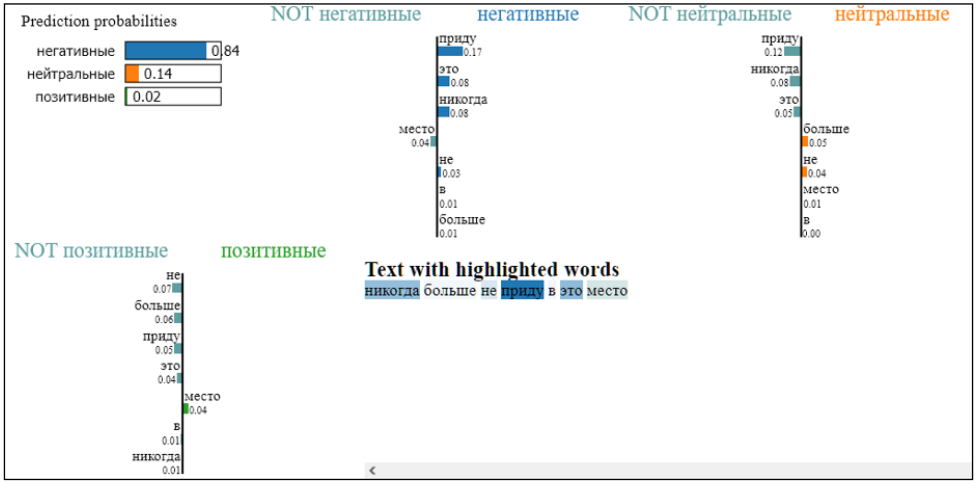

Рис. 15. Интерпретация результатов с помощью LIME

На рис. 15 можно увидеть благодаря весам каких слов данный текст был отмечен как негативный.

5. ЗАКЛЮЧЕНИЕ

Сравним эффективность моделей глубокого обучения с традиционными методами при решении задачи семантического анализа.

Таблица 3

Название модели	F1-мера	Время, в минутах
SVM+Tfidf	0.77	**< 1**
Logis+Tfidf	0.78	**< 1**
Catboost+Tfidf	0.76	3.5
NaiveBayes+Tfidf	0.75	**< 1**
CNN+word2vec	0.75	22
LSTM	0.80	2
ruBert_Base	0.26	300
rubert-tiny2	**0.83**	60
rubert-tiny2_one layer	0.81	10
ruGPT3small	0.79	4

Лучшие результаты среди классических методов показала логистическая регрессия: высокое качество прогнозирования и быстрая производительность.

Исследование подтверждает превосходство модели на основе BERT по качеству в сравнении с другими методами обучения. Однако, обучение этой модели требует времени. Уменьшение числа слоев энкодера может ускорить работу без существенной потери качества.

В дальнейших исследованиях в области семантического анализа текста рекомендуется уделить внимание получению эмбеддингов с использованием трансформера BERT и их последующему применению в сочетании с рекуррентными нейронными сетями, которые продемонстрировали достаточно высокое качество по сравнению с другими методами машинного и глубокого обучения.

Литература

1. Anjomshoaa H., Snagui Moharer R., Shirazi M. The effectiveness of training based on neuro--linguistic programming and cognitive-behavioral approach on students' anxiety, depression, and stress // International Journal of Pediatrics. 2021. Vol. 9. P. 14856–14866. DOI: 10.22038/IJP.2021.57871.4539
2. Begum A.J., Paulraj I.J.M., Banu S.H. (2022). Neuro-linguistic programming (NLP) is a promising communicative English language teaching technique // Sch Int J Linguist Lit. Vol. 5. P. 100–104. DOI:10.36348/sijll.2022.v05i03.004
3. Corcoran, C.M., Mittal, V.A., Bearden, C.E., E. Gur, R., Hitczenko, K., Bilgrami, Z., Savic, A., Cecchi, G.A., Wolff, P. Language as a biomarker for psychosis: a natural language processing approach // Schizophr. 2020. Vol. 226. P. 158–166. DOI: 10.1016/j.schres.2020.04.032
4. Chengyi Z., Brian Z.H., Andranik A.A., Beth C., Natural Language Processing to Identify Pulmonary Nodules and Extract Nodule Characteristics from Radiology Reports // Chest. 2021. Vol. 160. № 4. P. 1902–1914. DOI: 10.1016/j.chest.2021.05.048

5. Medhat W, Hassan A, Korashy H. Sentiment analysis algorithms and applications: a survey // Ain Shams Engineering Journal. 2014. Vol. 5. № 1. P. 1093–1113. DOI: 10.1016/j.asej.2014.04.011
6. Onan A. Sentiment analysis on product reviews based on weighted word embeddings and deep neural networks // Concurrency Comput. Pract. 2020. Vol. 33. P. 814–833. DOI: 10.1002/cpe.5909
7. Karthik R.V., Ganapathy S. A fuzzy recommendation system for predicting customers' interests using sentiment analysis and ontology in e-commerce // Applied Soft Computing. 2021. Vol. 108. DOI:10.1016/j.asoc.2021.107396
8. Bose R., Dey R.K., Roy S., Sarddar D. Sentiment analysis on online product reviews // Information and Communication Technology for Sustainable Development. 2020. Vol.993 P. 559–569. DOI:10.1007/978-981-13-7166-0_56
9. Mai L, Le B Joint sentence and aspect-level sentiment analysis of product comments // Annals of Operations Research. 2021. Vol. 300. № 2. P. 493–513. DOI: 10.1007/s10479-020-03534-7
10. Zhang Y., Du J., Ma X., Wen H., Fortino G. Aspect-based sentiment analysis for user reviews // Cognitive Computation. 2021. Vol. 13. № 5. P. 1114–1127. DOI: 10.1007/s12559-021-09855-4
11. Zellig S.H. Distributional Structure // Word. 2015 Vol. 10. № 2. P. 146–162. DOI: 10.1080/00437956.1954.11659520
12. Li Y., Yang T. Word Embedding for Understanding Natural Language: A Survey. In: Srinivasan S. (eds) Guide to Big Data Applications. Studies in Big Data. 2017. Vol 26. DOI: 10.1007/978-3-319-53817-4_4
13. Sparck K.J. A statistical interpretation of term specificity and its application in retrieval // Journal of Documentation. 1972. Vol. 28 № 1. P. 11–21. DOI: 10.1108/eb026526
14. Mikolov T., Chen K., Corrado G., Dean J. Efficient Estimation of Word Representations in Vector Space // arXiv preprint arXiv: 1301.3781.2013
15. Mikolov T., Quoc V. Le, Sutskever I. Exploiting Similarities among Languages for Machine Translation // arXiv preprint arXiv: 1309.4168v1.2013
16. Dorogush A.V., Ershov V., Gulin A. CatBoost: gradient boosting with categorical features support // Proceedings of the 32nd International Conference on Neural Information Processing Systems. 2018. P. 6639–6649
17. LeCun Y., Boser B., Denker J.S., Henderson D., Howard R.E., Hubbard W., Jackel L.D. Backpropagation Applied to Handwritten Zip Code Recognition. *// Neural Computation.* 1989. Vol. 1. № 4. P. 541–551. DOI: 10.1162/neco.1989.1.4.541
18. Hochreiter, S., Schmidhuber, J. Long short-term memory // Neural *Computation*. 1997. Vol. 9. № 8. P. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735
19. Vaswani A., Shazeer N., Parmar N., J. Uszkoreit, L. Jones, A. Gomez, A. Kaiser, I. Polosukhin. Advances in neural information processing systems. 2017. P. 5998–6008
20. Liu H., Li Z., Hall D., Liang P., Ma T. Sophia: A Scalable Stochastic Second-order Optimizer for Language Model Pre-training/ arXiv preprint arXiv: 2305.14342. 2023. URL: https://arxiv.org/abs/2305.14342

Semantic Analysis of Reviews About Organizations Using Machine Learning Methods

Evgeniy N. Platonov*
Moscow Aviation Institute (National Research University) Moscow, Russia
ORCID: https://orcid.org/0000-0001-8502-1350
e-mail: en.platonov@gmail.com

Irina R. Martynova**
Moscow Aviation Institute (National Research University) Moscow, Russia
ORCID: https://orcid.org/0009-0007-3140-2490
e-mail: irina.mart.r@gmail.com

Semantic analysis of organizational reviews is a key tool for assessing customer satisfaction levels. Business entities should regularly conduct analysis and emotional sentiment investigation to delve deeper into the data and gain a more comprehensive understanding of their operations, including through the use of machine learning methods. Presently, deep learning-based methods are garnering increased attention due to their high efficiency. In this study, we will focus on sentiment analysis tasks. To perform sentiment analysis, we will employ machine learning methods, including various approaches to text vectorization, deep learning models, and natural language processing (NLP) algorithms.

Keywords: natural language processing, classification task, gradient boosting, recurrent neural networks, convolutional neural networks, BERT, GPT.

For citation:
Platonov E.N., Martynova I.R. Semantic Analysis of Reviews About Organizations Using Machine Learning Methods. *Modelirovanie i analiz dannykh = Modelling and Data Analysis,* 2024. Vol. 14, no. 1, pp. 7–26. DOI: https://doi.org/10.17759/mda.2024140101 (In Russ., abstr. in Engl.).

References

1. Anjomshoaa H., Snagui Moharer R., Shirazi M. The effectiveness of training based on neuro--linguistic programming and cognitive-behavioral approach on students' anxiety, depression, and stress // International Journal of Pediatrics. 2021. Vol. 9. P. 14856–14866. DOI: 10.22038/IJP.2021.57871.4539

Evgeniy N. Platonov, PhD (Physics and Mathematics), Assistant Professor, Moscow Aviation Institute (National Research University), Moscow, Russia, ORCID: https://orcid.org/0000-0001-8502-1350, e-mail: en.platonov@gmail.com

****Irina R. Martynova,*** Student of the Institute of Information Technologies and Applied Mathematics, Moscow Aviation Institute (National Research University), Moscow, Russia, https://orcid.org/0009-0007-3140-2490, e-mail: irina.mart.r@gmail.com

2. Begum A.J., Paulraj I. J. M., Banu S.H. (2022). Neuro-linguistic programming (NLP) is a promising communicative English language teaching technique // Sch Int J Linguist Lit. Vol. 5. P. 100–104. DOI:10.36348/sijll.2022.v05i03.004
3. Corcoran, C.M., Mittal, V.A., Bearden, C.E., E. Gur, R., Hitczenko, K., Bilgrami, Z., Savic, A., Cecchi, G.A., Wolff, P. Language as a biomarker for psychosis: a natural language processing approach // Schizophr. 2020. Vol. 226. P. 158–166. DOI: 10.1016/j.schres.2020.04.032
4. Chengyi Z., Brian Z.H., Andranik A.A., Beth C., Natural Language Processing to Identify Pulmonary Nodules and Extract Nodule Characteristics from Radiology Reports // Chest. 2021. Vol. 160. № 4. P. 1902–1914. DOI: 10.1016/j.chest.2021.05.048
5. Medhat W, Hassan A, Korashy H. Sentiment analysis algorithms and applications: a survey // Ain Shams Engineering Journal. 2014. Vol. 5. № 1. P. 1093–1113. DOI: 10.1016/j.asej.2014.04.011
6. Onan A. Sentiment analysis on product reviews based on weighted word embeddings and deep neural networks // Concurrency Comput. Pract. 2020. Vol. 33. P. 814–833. DOI: 10.1002/cpe.5909
7. Karthik R.V., Ganapathy S. A fuzzy recommendation system for predicting customers' interests using sentiment analysis and ontology in e-commerce // Applied Soft Computing. 2021. Vol. 108. DOI:10.1016/j.asoc.2021.107396
8. Bose R., Dey R.K., Roy S., Sarddar D. Sentiment analysis on online product reviews // Information and Communication Technology for Sustainable Development. 2020. Vol.993 P. 559–569. DOI:10.1007/978-981-13-7166-0_56
9. Mai L, Le B Joint sentence and aspect-level sentiment analysis of product comments // Annals of Operations Research. 2021. Vol. 300. № 2. P. 493–513. DOI: 10.1007/s10479-020-03534-7
10. Zhang Y., Du J., Ma X., Wen H., Fortino G. Aspect-based sentiment analysis for user reviews // Cognitive Computation. 2021. Vol. 13. № 5. P. 1114–1127. DOI: 10.1007/s12559-021-09855-4
11. Zellig S.H. Distributional Structure // Word. 2015 Vol. 10. № 2. P. 146–162. DOI: 10.1080/00437956.1954.11659520
12. Li Y., Yang T. Word Embedding for Understanding Natural Language: A Survey. In: Srinivasan S. (eds) Guide to Big Data Applications. Studies in Big Data. 2017. Vol 26. DOI: 10.1007/978-3-319-53817-4_4
13. Sparck K.J. A statistical interpretation of term specificity and its application in retrieval // Journal of Documentation. 1972. Vol. 28 № 1. P. 11–21. DOI: 10.1108/eb026526
14. Mikolov T., Chen K., Corrado G., Dean J. Efficient Estimation of Word Representations in Vector Space // arXiv preprint arXiv: 1301.3781.2013
15. Mikolov T., Quoc V. Le, Sutskever I. Exploiting Similarities among Languages for Machine Translation // arXiv preprint arXiv: 1309.4168v1.2013
16. Dorogush A.V., Ershov V., Gulin A. CatBoost: gradient boosting with categorical features support // Proceedings of the 32nd International Conference on Neural Information Processing Systems. 2018. P. 6639–6649
17. LeCun Y., Boser B., Denker J.S., Henderson D., Howard R.E., Hubbard W., Jackel L.D. Backpropagation Applied to Handwritten Zip Code Recognition. *// Neural Computation.* 1989. Vol. 1. № 4. P. 541–551. DOI: 10.1162/neco.1989.1.4.541
18. Hochreiter, S., Schmidhuber, J. Long short-term memory // Neural *Computation*. 1997. Vol. 9. № 8. P. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735
19. Vaswani A., Shazeer N., Parmar N., J. Uszkoreit, L. Jones, A. Gomez, A. Kaiser, I. Polosukhin. Advances in neural information processing systems. 2017. P. 5998–6008
20. Liu H., Li Z., Hall D., Liang P., Ma T. Sophia: A Scalable Stochastic Second-order Optimizer for Language Model Pre-training/ arXiv preprint arXiv: 2305.14342. 2023. URL: https://arxiv.org/abs/2305.14342

Получена 26.02.2024 Received 26.02.2024
Принята в печать 04.03.2024 Accepted 04.03.2024