ВЕСТНИК
НОВОСИБИРСКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА

Научный журнал
Основан в ноябре 1999 года

Серия: Информационные технологии 2024. Том 22, № 3

СОДЕРЖАНИЕ

Бутенко Ю. И. Метод извлечения многокомпонентных терминологических единиц с правыми определениями из научно-технических текстов 5

Винокурова Д. В. Выбор оптимального языка программирования для генерации математических задач 15

Комлев Д. А. Дообучение модели CodeBERT для написания комментариев к SQL-запросам 28

Литвинов В. С., Власов А. А., Тейтельбаум Д. В. Программно-аппаратные решения потоковой обработки данных для компенсации температурных дрейфов скважинного инклинометра «Луч» 40

Швенк М. В., Бручес Е. П., Леман А. Я. Сравнение методов машинного обучения для решения задачи анализа тональности 49

Информация для авторов 62

ISSN 1818-7900 (Print). ISSN 2410-0420 (Online)
Вестник НГУ. Серия: Информационные технологии. 2024. Том 22, № 3
Vestnik NSU. Series: Information Technologies, 2024, vol. 22, no. 3

V E S T N I K
NOVOSIBIRSK STATE UNIVERSITY

Scientific Journal
Since 1999, November
In Russian

Series: Information Technologies 2024. Volume 22, № 3

CONTENTS

Butenko I. I. Method for extracting multi-component terminological units with right definitions from scientific and technical texts 5

Vinokurova D. V. Choosing the optimal programming language for the generation of mathematical problems 15

Komlev D. A. Further training of the CodeBERT model for writing comments on SQL queries 28

Litvinov V. S., Vlasov A. A., Teytelbaum D. V. Software and hardware solution for stream processing of data for compensation of temperature drifts of LWD orientation sensor «Looch» 40

Shvenk M. V., Bruches E. P., Leman A. Y. Comparison of machine learning methods for sentiment analysis 49

Instructions for Contributors..... 62

ISSN 1818-7900 (Print). ISSN 2410-0420 (Online)
Вестник НГУ. Серия: Информационные технологии. 2024. Том 22, № 3
Vestnik NSU. Series: Information Technologies, 2024, vol. 22, no. 3

The journal is published quarterly in Russian since 1999
by Novosibirsk State University Press

The address for correspondence
Faculty of Information Technologies, Novosibirsk State University
1 Pirogov Street, Novosibirsk, 630090, Russia
Tel. +7 (383) 363 42 46

E-mail address: inftech@vestnik.nsu.ru
On-line version: http://elibrary.ru

ISSN 1818-7900 (Print). ISSN 2410-0420 (Online)
Вестник НГУ. Серия: Информационные технологии. 2024. Том 22, № 3
Vestnik NSU. Series: Information Technologies, 2024, vol. 22, no. 3

Научная статья

УДК 004.89
DOI 10.25205/1818-7900-2024-22-3-5-14

Метод извлечения многокомпонентных терминологических единиц с правыми определениями из научно-технических текстов

Юлия Ивановна Бутенко

Московский государственный технический университет им. Н. Э. Баумана
Москва, Россия

iubutenko@bmstu.ru; https://orcid.org/0000-0002-9776-5709

Аннотация

В статье предложен метод извлечения русскоязычных многокомпонентных терминов, в структуре которых есть правые определения. Проведен анализ современных методов и программных средств извлечения специальной терминологии, а на его основе показано, что они охватывают термины только с левыми определениями. Исследована формальная структура многокомпонентных терминологических единиц с правыми определениями, где особое внимание уделено их грамматическим особенностям. Обоснована нецелесообразность применения лемматизации ко всем компонентам термина. Проанализирована корректность работы морфологических анализаторов в аспекте их применимости к извлечению многокомпонентных терминов. Приведены модели пятикомпонентных терминов, которые стали основой для разработки метода извлечения русскоязычных многокомпонентных терминов с правыми определениями. В моделях определены ядерный элемент, левое и правое определения, а также грамматические признаки правого определения. Проиллюстрированы различия в списках терминов-кандидатов при использовании традиционных подходов, использующих лемматизацию на первом этапе, и предложенного метода извлечения многокомпонентных терминов с правыми определениями.

Ключевые слова

многокомпонентный термин, структура термина, модель термина, ядерный элемент, многокомпонентный термин с правыми определениями, лемматизация

Для цитирования

Бутенко Ю. И. Метод извлечения многокомпонентных терминологических единиц с правыми определениями из научно-технических текстов // Вестник НГУ. Серия: Информационные технологии. 2024. Т. 22, № 3. С. 5–14. DOI 10.25205/1818-7900-2024-22-3-5-14

Method for Extracting Multi-Component Terminological Units with Right Definitions from Scientific and Technical Texts

Iuliia I. Butenko

Bauman Moscow State Technical University, Moscow, Russian Federation

iubutenko@bmstu.ru; https://orcid.org/0000-0002-9776-5709

Abstract

The paper proposes a method for extracting Russian-language multicomponent terms with right definitions in their structure. The analysis of modern methods, techniques and software tools for extraction of special terminology is carried out, and on its basis it is shown that they cover terms only with left definitions only. The formal structure of Russian-language multi-component terminological units with right definitions is investigated, where special attention is paid to their

grammatical features, which include gender, case, number for Russian language nouns and adjectives. The inexpediency of applying lemmatisation to all components of a term is substantiated. The correctness of morphological analyzers of Russian texts is analyzed in the aspect of their applicability to the extraction of multi-component terms. The models of five-component terms are given, which became the basis for the development of the method of extraction of Russian-language multicomponent terms with right definitions. The proposed structural models identify the nuclear element, left and right definitions, and grammatical features of the right definition for Russian-language multicomponent terms. The paper also illustrates he differences in the lists of Russian-language candidate terms when using traditional approaches that use lemmatisation at the first stage and the proposed method for extraction of multicomponent terms with right definitions.

Keywords

multicomponent term, term structure, term model, nuclear element, multicomponent term with right definitions, lemmatisation

For citation

Butenko Iu. I. Method for extracting multi-component terminological units with right definitions from scientific and technical texts. *Vestnik NSU. Series: Information Technologies*, 2024, vol. 22, no. 3, pp. 5–14 (in Russ.) DOI 10.25205/1818-7900-2024-22-3-5-14

Введение

К настоящему времени разработано множество методов автоматического извлечения терминов, и число работ новых работ в последние годы только увеличивается [1; 2]. Общая схема для большинства методов извлечения терминов имеет следующий вид: сбор кандидатов: фильтрация слов и словосочетаний, извлекаемых из коллекции документов, по статистическим или лингвистическим принципам; подсчет признаков: перевод каждого кандидата в вектор признакового пространства; вывод на основе признаков: оценка вероятности быть термином для каждого кандидата на основе значений признаков. С целью снижения шума при сборе терминов кандидатов производится дополнительная фильтрация: по частоте, по содержанию в составе термина-кандидата стоп-слов, по длине или содержанию в термине-кандидате особых символов [3]. Также существует множество схем классификации методов извлечения терминологии из текстов [4]. Традиционно среди наиболее распространенных методов выявления терминов в текстах используют лингвистические и статистические методы, а также методы, основанные на машинном обучении и использовании различных информационных ресурсов [1].

Анализ групп методов извлечения терминов показал типовых ряд ограничений. Так, статистические методы, будучи независимыми от языка, извлекают не все термины и их употребления в тексте [5]; методы на основе синтаксических шаблонов зависят от языка, кроме того, в них велика доля шума, так как модели терминов совпадают по форме с моделями общеупотребительных словосочетаний естественного языка [6; 7]. Кроме того, современные исследования в области терминоведения свидетельствуют об изменениях в способах образования терминов в русском языке [8]. Методы на основе машинного обучения требуют наличия большого количества размеченных данных [9; 10]. Потенциально такими ресурсами являются специальные корпуса, но их анализ показал, что у них или маленький объем размеченных данных, или существенные ограничения предметных областей текстов: чаще всего это тексты документов различных международных организаций. Кроме того, при анализе результатов обработки терминологии программными средствами также выявлена следующая особенность: чаще всего автоматически извлекают термины, состоящие из 2–3 элементов с левыми определениями в своей структуре. Анализ работ по терминоведению (табл. 1) свидетельствует о том, что среди 3–6-компонентных терминов, которые в общем числе терминов составляют порядка 35–40 % в зависимости от предметной области, в своей структуре содержат правые определения [11; 12].

Таблица 1

Количество компонентов в русскоязычных терминах

Table 1

Amount of components in Russian terms

Количество компонентов	1	2	3	4	5	6 и более
%	35	28	25	7	3	2

Таким образом, целью статьи является разработка метода извлечения многокомпонентных терминологических единиц с правыми определениями из научно-технических текстов.

Лингвистические особенности формальной структуры многокомпонентных терминов

При анализе предложения «В системах поддержки принятия решений используются информационный поиск, интеллектуальный анализ данных, имитационное моделирование, нейронные сети, когнитивное моделирование и др.», где сначала выполнятся лемматизация, а затем применяются синтаксические шаблоны, получается следующий список терминов-кандидатов: *система, поддержка, принятие, решение, информационный поиск, интеллектуальный анализ, данные, имитационный моделирование нейронный сеть, когнитивный моделирование*. Если проанализировать каждый термин-кандидат в этом списке, а также сравнить его с зафиксированными терминами в словарях, то правильность их извлечения не вызывает никаких сомнений. Однако на практике встречаются многокомпонентные термины, которые в свою структуру включают терминоэлементы, которые сами по себе являются терминами. Обобщенная модель многокомпонентной терминологической единицы представлена на рис. 1.

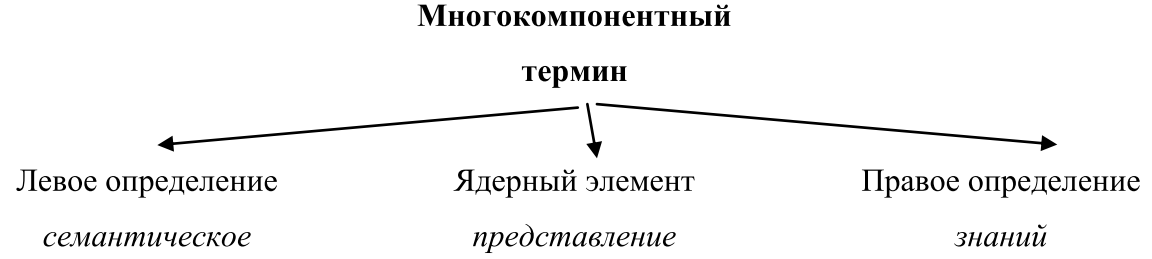

Рис. 1. Обобщенная модель многокомпонентного термина
Fig. 1. Generalized model of a multi-component term

В структуре термина лингвисты выделяют ядерный элемент, левые и правые определения. Ядерный элемент – главный элемент в структуре многокомпонентного термина, который вступает в словоизменительную парадигму в предложении. Левое и правое определение уточняют значение ядерного элемента. При этом левое определение чаще всего выражено именами прилагательными, которые наследуют грамматические признаки рода, числа и падежа имени существительного, перед которым стоят, а правые определения выражены именными группами, которые стоят после ядерного элемента, и при этом их грамматические характеристики остаются неизменными.

При реализации традиционных методов извлечения терминов на первом этапе проводится лемматизация, суть которой состоит в приведении всех слов в начальную форму. Такой подход приводит к тому, что правые определения выделяются как отдельные термины. На рис. 2 по-

казана схема наследования грамматических характеристик элементов термина при его нормализации, где стрелками показаны грамматические характеристики, которые должны совпадать с терминологической единицей при корректном извлечении терминов с правыми определениями.

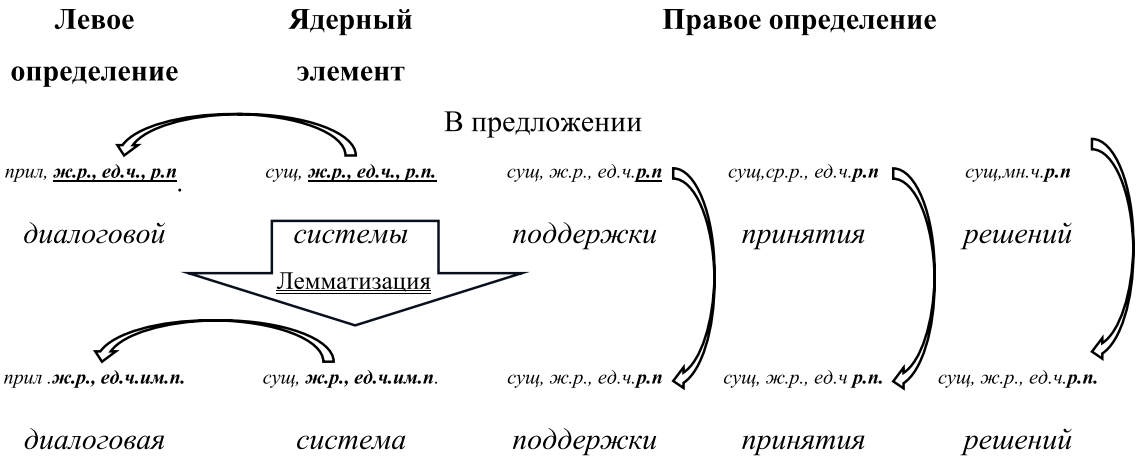

Рис. 2. Схема наследования грамматических характеристик при нормализации многокомпонентного термина
Fig. 2. Scheme of grammatical characteristics in normalisation of a multi-component term

Таким образом, для повышения эффективности методов извлечения многокомпонентных терминов за счет обработки терминов с правыми определениями необходимо разработать специальный подход, учитывающий их морфологические особенности.

Морфологический анализ как этап процедуры извлечения многокомпонентных терминов

В настоящее время морфологические анализаторы широко используются для обработки текстов на разных языках, а их эффективность в значительной степени зависит от грамматического строя самого языка. Для русского языка информация о морфологических особенностях слова может быть разделена на четыре группы помет: лексема, множество ее грамматических признаков, множество грамматических признаков словоформы и нестандартная информация грамматической формы. Лексемой является начальная форма слова, которая используется в словарях, а также указывается частеречная принадлежность самой лексемы. Вторая группа помет отражает грамматические признаки лексемы, такие как род у имени существительного, разряд у имен прилагательных, переходность у глагола, при этом в третьей группе содержатся грамматические признаки для словоформы: число и падеж у имени существительного, время, лицо, число у глаголов настоящего и будущего времени, род и число у глаголов прошедшего времени. Четвертая группа носит факультативный характер и может отражать информацию о диалектных особенностях написания слова или его нестандартных грамматических формах.

В аспекте решение задачи извлечения многокомпонентных терминов с правыми определениями качество морфологической разметки имеет первостепенное значение. Частеречный анализ многокомпонентных терминов показал, что они имеют ряд ограничений на используемые для их образования лексические единицы. Так, в образовании многокомпонентных терминов не используются местоимения, глаголы, междометия, а также знаки препинания. Такие части речи, как имя существительное, имя прилагательное, имя числительное, наречие, предлог, при-

частие чаще всего входят в состав многокомпонентных терминов, а ядерным элементом всегда выступает имя существительное [13].

Стоит отметить, что возможности морфологических анализаторов для русского языка достигли достаточно высокого уровня развития, однако на практике встречаются некоторое ошибки, например:

1) Программа неверно определяет падеж существительных:

Когда^{СОЮЗ} *потребитель*$^{СУЩ,од,мр\ ед,им}$ *хочет*^{ГЛ,несов,перех,ед,3л,наст,изъяв} *сохранить*^{ИНФ,сов,перех} *анонимность*$^{СУЩ,неод,жр\ ед,им}$ (распознан именительный падеж вместо винительного);

2) Система не распознает сокращения:

Слушатели$^{СУЩ,од,мр\ мн,им}$ *радиоканала*$^{СУЩ,неод,мр\ ед,рд}$, *участники*$^{СУЩ,од,мр\ мн,им}$ *интернет*$^{СУЩ,неод,мр\ ед,им}$*-сообществ*$^{СУЩ,неод,ср\ мн,рд}$ *и*^{СОЮЗ} *др*^{НЕИЗВ} (сокращение «др.» не распознается анализатором)

3) Неверно определяется часть речи, например:

Такая$^{ПРИЛ,мест\text{-}п\ жр,ед,им}$ *оценка*$^{СУЩ,неод,жр\ ед,им}$ *должна*$^{КР\_ПРИЛ\ жр,ед}$ *появиться*^{ИНФ,сов,неперех} *в*^{ПР} *результате*$^{СУЩ,неод,мр\ ед,пр}$ *проведения*$^{СУЩ,неод,ср\ ед,рд}$ *соответствующего*$^{ПРИЛ\ мр,ед,рд}$ *маркетингового*$^{ПРИЛ\ мр,ед,рд}$ *либо*^{СОЮЗ} *иного*$^{ПРИЛ,мест\text{-}п\ ср,ед,рд}$ *исследования*$^{СУЩ,неод,ср\ ед,рд}$ (местоимение «такая» определяется как имя прилагательное)

4) Составные части речи рассматриваются как отдельные единицы:

Обновление$^{СУЩ,неод,ср\ ед,им}$ *информации*$^{СУЩ,неод,жр\ ед,рд}$ *в*^{ПР} *печатных*$^{ПРИЛ,кач\ мн,рд}$ *СМИ*$^{СУЩ,неод,хр,pl,0\ мн,им}$ *не*^{ЧАСТ} *может*$^{ГЛ,несов,неперех\ ед,3л,наст,изъяв}$ *успеть*^{ИНФ,сов,неперех} *ни*^{ЧАСТ} *за*^{ПР} *телевидением*$^{СУЩ,неод,ср\ ед,тв,}$ *ни*^{ЧАСТ} *тем*^{СОЮЗ} *более*^{Н} *за*^{ПР} *Интернетом*$^{СУЩ,неод,мр\ ед,тв}$; (составной союз «тем более» распознан как две единицы)

Данные особенности должны быть учтены при разработке моделей и методов извлечения многокомпонентных терминов из научно-технических текстов.

Метод извлечения многокомпонентных терминов с правыми определениями

Многокомпонентные термины с правыми определениями по своей структуре чаще всего состоят их 3–6 компонентов. Терминологи также выделяют и терминологические единицы большей длины, однако на практике они со временем времени уменьшают число компонентов из-за своей громоздкости. В табл. 2 приведены модели пятикомпонентных терминов с указанием ядерного элемента, а также грамматических характеристик правых определений. По результатам анализа работы морфологических анализаторов для имен прилагательных учитывается только частеречная принадлежность, а причастия и деепричастия не использованы, так как распознаются именами прилагательными.

Таблица 2

Структурные модели 5-компонентных терминологических единиц

Table 2

Structural models of 5 component terminological units

	Количество компонентов	**Модель**
1	**2**	**3**
1	5	Имя прилагательное + имя существительное (ядерный элемент) + имя прилагательное + имя существительное в творительном падеже (с предлогом «с»)
2	5	Имя прилагательное + имя существительное (ядерный элемент) + имя прилагательное + имя существительное в предложном падеже

Окончание табл. 2

1	2	3
3	5	Имя прилагательное + имя существительное (ядерный элемент) + имя существительное в предложном падеже + имя существительное в родительном падеже
4	5	Имя прилагательное + имя существительное (ядерный элемент) + имя существительное в творительном падеже (с предлогом) + имя существительное в родительном падеже
5	5	Имя прилагательное + имя прилагательное + существительное (ядерный элемент) + существительное в творительном падеже (с предлогом)
6	5	Имя прилагательное + имя существительное (ядерный элемент) + имя существительное в творительном падеже + имя существительное в творительном падеже (с предлогом)
7	5	Имя прилагательное + имя прилагательное + имя существительное (ядерный элемент) + (прилагательное + имя существительное в творительном падеже)
8	5	Имя прилагательное + имя существительное (ядерный элемент) + (имя прилагательное + имя прилагательное + имя существительное в творительном падеже)
9	5	Имя прилагательное + имя существительное (ядерный элемент) + (имя существительное в творительном падеже + имя существительное в родительном падеже + имя существительное в родительном падеже)

Предлагаемый метод к автоматическому извлечению русскоязычных многокомпонентных терминов на основе структурных моделей англо- и русскоязычных терминологических словосочетаний состоит из этапов, представленных на рис. 3.

Рис. 3. Этапы метода извлечения русскоязычных многокомпонентных терминов
Fig. 3. Stages of the method for extracting Russian multicomponent terms

Работу предложенного метода по извлечению многокомпонентных терминов с правыми определениями можно проиллюстрировать на следующем примере.

В системах поддержки принятия решений используются информационный поиск, интеллектуальный анализ данных, имитационное моделирование, нейронные сети, когнитивное моделирование и др.

1. С помощью морфологического анализатора PyMorphy2 к каждой лексической единице приписывается ее морфологические характеристики:

В ПР системах СУЩ,неод,жр мн,пр поддержки СУЩ,неод,жр ед,рд принятия СУЩ,неод,ср ед,рд решений СУЩ,неод,ср мн,рд используются ГЛ,несов,неперех мн,3л,наст,изъяв информационный ПРИЛ мр,ед,им поиск СУЩ,неод,мр ед,им ,ЗПР интеллектуальный ПРИЛ,кач мр,ед,им анализ СУЩ,неод,мр ед,им данных СУЩ,неод,хр,pl мн,рд , ЗПР имитационное ПРИЛ ср,ед,им моделирование СУЩ,неод,ср ед,им, ЗПР нейронные ПРИЛ мн,им сети СУЩ,неод,жр ед,рд, когнитивное ПРИЛ *ср,ед,им моделирование СУЩ,неод,ср ед,им и СОЮЗ др СУЩ,неод,мр,0,аббр ед,им*.

2. На основе частеричного анализа структурных элементов многокомпонентных терминов из текста исключаем глаголы, союзы, местоимения, частицы, а также знаки препинания.

~~В ПР~~ системах СУЩ,неод,жр мн,пр поддержки СУЩ,неод,жр ед,рд принятия СУЩ,неод,ср ед,рд решений СУЩ,неод,ср мн,рд ~~используются~~ ~~ГЛ,несов,неперех мн,3л,наст,изъяв~~ информационный ПРИЛ мр,ед,им поиск СУЩ,неод,мр ед,им ,~~ЗПР~~ интеллектуальный ПРИЛ,кач мр,ед,им анализ СУЩ,неод,мр ед,им данных СУЩ,неод,хр,pl мн,рд , ~~ЗПР~~ имитационное ПРИЛ ср,ед,им моделирование СУЩ,неод,ср ед,им, ЗПР нейронные ПРИЛ мн,им сети СУЩ,неод,жр ед,рд, когнитивное ПРИЛ ср,ед,им моделирование СУЩ,неод,ср ед,им ~~и СОЮЗ др СУЩ,неод,мр,0,аббр ед,им~~.

Таким образом, в списке терминов-кандидатов остаются следующие цепочки слов:

системах СУЩ,неод,жр мн,пр *поддержки* СУЩ,неод,жр ед,рд *принятия* СУЩ,неод,ср ед,рд *решений* СУЩ,неод,ср мн,рд

информационный ПРИЛ мр,ед,им *поиск* СУЩ,неод,мр ед,им

интеллектуальный ПРИЛ,кач мр,ед,им *анализ* СУЩ,неод,мр ед,им *данных* СУЩ,неод,хр,pl мн,рд

имитационное ПРИЛ ср,ед,им *моделирование* СУЩ,неод,ср ед,им

нейронные ПРИЛ мн,им *сети* СУЩ,неод,жр ед,рд,

когнитивное ПРИЛ ср,ед,им *моделирование* СУЩ,неод,ср ед,им

3. На третьем этапе полученные цепочки слов необходимо проверить на наличие «стоп-слов», которые по морфологическим параметрам могут подходить под синтаксические шаблоны многокомпонентных терминов, но при этом образуют свободные словосочетания со структурой «нетермин + термин», например, *современная* технология, *анализируемый* подход.

4. Полученные цепочки слов соотносим с шаблонами терминологических словосочетаний, имеющихся в базе структурных моделей терминов. Ядерные элементы выделяем жирным, левые определения принимают грамматические признаки рода, числа и падежа такие же, как у ядерного элемента, а грамматические признаки правых определений остаются неизменными.

система СУЩ,неод,жр мн,пр *поддержки* СУЩ,неод,жр ед,рд *принятия* СУЩ,неод,ср ед,рд *решений* СУЩ,неод,ср мн,рд

информационный ПРИЛ мр,ед,им **поиск** СУЩ,неод,мр ед,им

интеллектуальный ПРИЛ,кач мр,ед,им **анализ** СУЩ,неод,мр ед,им *данных* СУЩ,неод,хр,pl мн,рд

имитационное ПРИЛ ср,ед,им **моделирование** СУЩ,неод,ср ед,им

нейронные ПРИЛ мн,им **сети** СУЩ,неод,жр ед,рд,

когнитивное ПРИЛ ср,ед,им **моделирование** СУЩ,неод,ср ед,им

Предложенный метод извлечения многокомпонентных терминов с правыми определениями позволяет повысить эффективность извлечения терминов за возможности обработки терминов большей длины за счет учета их правых определений. В рассмотренном примере традиционным способом было извлечено 10 терминов-кандидатов с максимальным числом компонентов – два, а с использованием предложенного метода – 6 и максимальным числом компонентов термина – 4.

Для оценки эффективности предложенного метода использовалась экспертная оценка, проведенная филологами. Всего проанализировано 20 текстов научно-технических статей по космонавтике, опубликованных в журнале «Космические исследования» в 2018–2019 гг. Оценка качества метода извлечения терминов (табл. 3) проводилась путем сравнения списков терминов, извлеченных системой, со списком филологов.

Таблица 3

Оценка качества метода извлечения терминов из научно-технических текстов, в %

Table 3

Quality assessment of a method for extracting terms from scientific and technical texts

	Полнота	**Точность**	**F-мера**
Извлечение терминов	91	75	82

Стоит отметить, что при оценке качества извлечения терминов из текстов наиболее спорными стали одно- и двухкомпонентные термины, у которых есть как терминологическое значение, так и общеупотребительное, например, лексическая единица игрок является термином из теории игр и общеупотребительным словом. При этом такое явление не наблюдалось при извлечении терминов из трех и более компонентов. Более того, подход с использованием синтаксических шаблонов многокомпонентных терминов позволит значительно расширить потенциал систем автоматической генерации текстов, так как такие модели позволяют определить в терминах те элементы, которые вступают в словоизменительную парадигму в предложении.

Заключение

Проведенный анализ методов и программных средств автоматического извлечения терминов показал, что наиболее эффективными являются методы извлечения одно-, двухкомпонентных терминов. Вместе с тем изучение работ современных терминологов показывает, что порядка 35–40 % терминологических единиц состоят из 3 и более элементов. В структуре термина выделены ядерный элемент, левые и правые определения, при этом левые определения имеют те же грамматические признаки рода, числа и падежа, что и ядерный элемент, а правые грамматические признаки правых определений остаются неизменными. Выделены модели пятикомпонентных терминов, в которых указаны ядерный элемент и грамматические характеристики правых определений. Анализ эффективности предложенного метода извлечения многокомпонентных терминов с правыми определениями проведен на фрагментах из научно-технических текстов. В основе метода заложены результаты морфологического анализа текста, по результатам которого из текста извлекаются части речи, не входящие в многокомпонентные термины, а получившиеся в результате цепочки слов сопоставляют с моделями терминов. При этом лемматизация предусмотрена только для ядерного элемента и левых определений, правые определения остаются неизменными. Предложенный метод позволяет повысить эффективность извлечения терминов из-за возможности обработки терминов большей длины за счет учета их правых определений, а усовершенствованные модели многокомпонентных терминов могут расширить потенциал систем автоматической генерации текстов при обработке специальной терминологии.

Список литературы

1. **Nugumanova A., Akhmed-Zaki D., Mansurova M., Baiburin Y., Maulit A.** NMF-based approach to automatic term extraction // Expert Systems with Applications. 2022. № 199. P. 117179. DOI: 10.1016/j.eswa.2022.117179

2. **Lossio-Ventura J. A., Jonquet C., Roche M. et al.** Biomedical term extraction: overview and a new methodology // Inf Retrieval. 2019. № 19. С. 59–99. DOI: 10.1007/s10791-015-9262-2
3. **Astrakhantsev N. A., Fedorenko D. G., Turdakov D. Y.** Methods for automatic term recognition in domain-specific text collections: A survey // Programming and Computer Software. 2015. Vol. 41, No. 6. P. 336–349. DOI 10.1134/S036176881506002X.
4. **Granado N. G., Drouin P., Picton A.** From statistical analysis to machine learning: Language in the service of terminology // Ela. Etudes de linguistique appliquee. 2022. № 208(4). P. 447-
5. **Клышинский Э. С., Кочеткова Н. А., Карпик О. В.** Метод выделения коллокаций с использованием степенного показателя в распределении Ципфа // Новые информационные технологии в автоматизированных системах. 2018. № 21. С. 220–225.
6. **Наместников А. М., Филлипов А. А., Шагабутдинов И. М.** Подход к извлечению многословных терминов из текстов на естественном языке с применением синтаксических шаблонов // Автоматизация процессов управления. 2021. № 3 (65). С. 87–95. DOI: 10.35752/1991-2927-2021-3-65-87-95
7. **Бутенко Ю. И., Строганов Ю. В., Сапожков А. М.** Метод извлечения русскоязычных многокомпонентных терминов в корпусе научно-технических текстов // Прикладная информатика. 2021. № 6. С. 21–27. DOI: 10.37791/2687-0649-2021-16-6-21-27
8. **Козловская Н. В., Янурик С.** ИИ-композиты как объект неологии и неографии в XXI веке // Филологические науки. Научные доклады высшей школы. 2021. № 2. С. 23–30. DOI: 10.20339/PhS.2-21.023
9. **Большакова Е. И., Лукашевич Н. В., Нокель М. А.** Извлечение однословных терминов из текстовых коллекций на основе методов машинного обучения // Информационные технологии. 2013. № 7. С. 31–36.
10. **Бручес Е. П., Батура Т. В.** Метод автоматического извлечения терминов из научных статей на основе слабоконтролируемого обучения // Вестник НГУ. Серия: Информационные технологии. 2021. Т. 19, № 2. С. 5–16. DOI: 10.25205/1818-7900-2021-19-2-5-16
11. **Гринев-Гриневич С. В., Сорокина Э. А., Молчанова М. А.** Терминоведение. Изд. 3-е, испр. и доп. М.: ЛЕНАРД, 2023. 500 с.
12. **Бутенко Ю. И., Николаева Н. С., Карцева Е. Ю.** Структурные модели англоязычных терминов для автоматической обработки корпусов научно-технических текстов // Вестник Российского университета дружбы народов. Серия: Теория языка. Семиотика. Семантика. 2022. Т. 14, № 1. С. 80–95 DOI: 10.22363/2313-2299-2022-13-1-80-95
13. **Бутенко Ю. И., Строганов Ю. В., Сапожков А. М.** Система извлечения многокомпонентных терминов и их переводных эквивалентов из параллельных научно-технических текстов // Научно-техническая информация: Серия 2. Информационные процессы и системы. 2022. № 9. С. 12–21. DOI: 10.36535/0548-0027-2022-09-3

References

1. **Nugumanova A., Akhmed-Zaki D., Mansurova M., Baiburin Y., Maulit A.** NMF-based approach to automatic term extraction. *Expert Systems with Applications*, 2022, no. 199, p. 117179. DOI: 10.1016/j.eswa.2022.117179
2. **Lossio-Ventura, J. A., Jonquet, C., Roche, M. et al.** Biomedical term extraction: overview and a new methodology. *Inf Retrieval*, 2019, no. 19, pp. 59–99. DOI: 10.1007/s10791-015-9262-2
3. **Astrakhantsev N. A., Fedorenko D. G., Turdakov D. Y.** Methods for automatic term recognition in domain-specific text collections: A survey. *Programming and Computer Software*, 2015, vol. 41, no. 6, pp. 336–349. DOI 10.1134/S036176881506002X.

4. **Granado N. G., Drouin P., Picton A.** From statistical analysis to machine learning: Language in the service of terminology. *Ela. Etudes de linguistique appliquee*, 2022, no. 208 (4), pp. 447–467. DOI: 10.3917/ela.208.0067
5. **Klyshinskij E. S., Kochetkova N. A., Karpik O. V.** Method of collocation extraction using the stepped index in the Zipf distribution. *Information technologies in automated systems*, 2018, no. 21, pp. 220–225. (in Russ.)
6. **Namestnikov A. M., Filippov A. A., Shigabutdinov I. M.** The extraction of terms consisting of several words from texts in natural languages using the syntactic patterns. *Automation of Control Processes*, 2021, no 3(65), p. 87–95. DOI: 10.35752/1991-2927-2021-3-65-87-95 (in Russ.)
7. **Butenko Iu. I., Stroganov Yu. V., Sapozhkov A. M.** Method for the extraction of russian language multicomponent terms from scientific and technical texts. *Applied Informatics*, 2021, no. 6, pp. 21–27. DOI: 10.37791/2687-0649-2021-16-6-21-27 (in Russ.)
8. **Kozlovskaya N. V., Janurik S. Z.** "II-composites" as an object of neology and neography of the xxi century. *Philological Sciences. Scientific essays of higher school*, 2021, no. 2, pp. 23–30. DOI: 10.20339/PhS.2-21.023 (in Russ.)
9. **Bolshakova E. I., Loukachevitch N. V., Nokel M. A.** Single-word term extraction from text collections based on machine learning. *Informacionnye Tehnologii* [*Information Tecjnologies*], 2013, no. 7, pp. 31–36. (in Russ.)
10. **Bruches E. P., Batura T. V.** Method for automatic term extraction from scientific articles based on weak supervision. Vestnik NSU. Series: *Information Technologies*, 2021, vol. 19, no. 2, pp. 5–16. DOI: 10.25205/1818-7900-2021-19-2-5-16 (in Russ.)
11. **Grinev-Grinevich S. V., Sorokina E. A., Molchanova M. A.** Terminovedenie. Moscow, LENARD, 2023, 500 p. (in Russ.)
12. **Butenko Iu. I., Nikolaeva N. S., Kartseva E. Yu.** Structural models of English terms of automated processing of scientific and technical texts corpora. *RUDN Journal of Language Studies, Semiotics and Semantics*, 2022, vol. 14, no. 1, pp. 80–95. DOI: 10.22363/2313-2299-2022-13-1-80-95 (in Russ.)
13. **Butenko Iu. I., Stroganov Yu. V., Sapozhkov A. M.** System for extracting multicomponent terms and their translated equivalents from parallel scientific and technical texts. *Nauchno-tekhnicheskaya informaciya. Seriya 2. Informacionnye processy i sistemy* [*Scientific and technical information. Series 2. Information processes and systems*], 2022, no. 9, pp. 12–21. DOI: 10.36535/0548-0027-2022-09-3 (in Russ.)

Сведения об авторе

Бутенко Юлия Ивановна, кандидат технических наук

Information about the Author

Iuliia I. Butenko, Candidate of Technical Sciences

Статья поступила в редакцию 23.04.2024;
одобрена после рецензирования 14.08.2024; принята к публикации 14.08.2024

The article was submitted 23.04.2024;
approved after reviewing 14.08.2024; accepted for publication 14.08.2024

Научная статья

УДК 167.2:004.432.2
DOI 10.25205/1818-7900-2024-22-3-15-27

Выбор оптимального языка программирования для генерации математических задач

Дарья Валентиновна Винокурова

Российский государственный педагогический университет им. А. И. Герцена
Санкт-Петербург, Россия

d.v.vinokurova@gmail.com, https://orcid.org/0000-0003-0892-1739

Аннотация

Сравниваются математические библиотеки языков веб-программирования JavaScript, PHP, Python для создания генераторов в области некоторых тем математического анализа и вычислительной математики. Основная цель исследования − провести эксперимент с заданным набором задач, используя библиотеки Math.js, Algebrite, Nerdamer, MathPHP, NumPy, SymPy, SciPy, чтобы определить оптимальные по функциональности и производительности для осуществления символьных и численных вычислений. Экспериментальное исследование выполнялось с помощью перечисленных библиотек, в которых осуществлялось вычисление соответствующих задач с измерением скорости их выполнения. Представлен сравнительный анализ результатов исследования. Показаны основные проблемы, которые возникали в ходе эксперимента в различных библиотеках. Полученные результаты могут быть использованы разработчиками и исследователями, которые занимаются проектированием и реализацией генераторов математических задач. В процессе работы выявлено, что библиотеки JavaScript и PHP в полной мере не поддерживают все функции для создания генераторов математических задач. Python оказался намного эффективнее как в символьных, так и в численных вычислениях.

Ключевые слова

разработка генераторов, JavaScript, PHP, Python, математические библиотеки, символьные вычисления, численные вычисления

Для цитирования

Винокурова Д. В. Выбор оптимального языка программирования для генерации математических задач // Вестник НГУ. Серия: Информационные технологии. 2024. Т. 22, № 3. С. 15–27. DOI 10.25205/1818-7900-2024-22-3-15-27

Choosing the Optimal Programming Language for the Generation of Mathematical Problems

Darya V. Vinokurova

Herzen State Pedagogical University of Russia,
St. Petersburg, Russian Federation

d.v.vinokurova@gmail.com, https://orcid.org/0000-0003-0892-1739

Abstract

This paper compares mathematical libraries of web programming languages JavaScript, PHP, Python to create generators in the field of some topics of mathematical analysis and computational mathematics. The main objective of the study is to conduct an experiment with a given set of tasks, using the libraries Math.js, Algebrite, Nerdamer, MathPHP, NumPy, SymPy, SciPy to determine the optimal functionality and performance for performing character and numerical

computing. The experimental study was carried out with the help of the libraries listed, where the corresponding tasks were computed with the measurement of their speed. A comparative analysis of the obtained results of the study is given. The main problems that arose during the experiment in different libraries are shown. The obtained results can be used by developers and researchers who are involved in the design and implementation of generators of mathematical problems. In the process of work it is identified that JavaScript and PHP libraries do not fully support all functions for creating generators of mathematical problems. Python was much more efficient in both symbolic and numerical calculations.

Keywords

Generator development, JavaScript, PHP, Python, mathematical libraries, symbolic computation, numerical computation

For citation

Vinokurova D. V. Choosing the optimal programming language for the generation of mathematical problems. *Vestnik NSU. Series: Information Technologies*, 2024, vol. 22, no. 3, pp. 15–27 (in Russ.) DOI 10.25205/1818-7900-2024-22-3-15-27

Введение

Генерация математических задач всегда являлась актуальной проблемой. Период пандемии 2020–2021 гг. заставил в большей степени задуматься над тем, каким образом повысить уникальность задач и автоматизировать процесс их проверки. Многовариантные задачи во многих печатных или электронных изданиях задачников создаются на основе определенного шаблона, в котором меняются параметры или функции. Типовые варианты используются многократно, и возникает проблема, связанная с их обновлением. Выпуск новых задачников происходит не так часто, поскольку требуются значительные усилия, чтобы их разработать и проверить корректность. В данном случае облегчить трудности способна генерация задач, которая осуществляется путем использования языка программирования, более подходящего для конкретной области. Наличие шаблона, под который можно сгенерировать задачи, позволяет вносить необходимые изменения, что обеспечивает увеличение разнообразия задач и адаптацию под необходимые потребности. Правильные ответы к сгенерированным задачам предоставляют мгновенную обратную связь.

Под программной генерацией задач будем понимать создание множества вариантов заданного класса, исключая повторяющиеся наборы, что обеспечивается соответствующим алгоритмом [1]. Под задачей подразумевается математическая задача по выбранной дисциплине определенного типа: из математического анализа пределы, производные, интегралы, из вычислительной математики нелинейные уравнения, численное интегрирование, интерполирование функций и системы линейных алгебраических уравнений, которые необходимо решить, применяя определенные функции языков программирования. Генерация задач требует наличия различных параметров в условиях, что обеспечивает разнообразие и предоставляет преподавателю возможность получать различные комплекты вариантов.

Рассмотрим ряд математических тренажеров на английском языке – MathsBot.com, Wolfram Problem Generator, Algebramaker.com, которые позволяют генерировать различные виды задач.

Приложение MathsBot.com содержит спектр задач по арифметике, алгебре, геометрии, статистике, которые посвящены элементарным математическим знаниям, не включающим математические дисциплины высшей школы. Существует возможность генерировать дифференцированные вопросы по уровням сложности, вопросы вида «пример-проблема», а также вопросы из различных тем, подобно материалам учебников и рабочих тетрадей.

Wolfram Generator[1] предоставляет инструмент для практической отработки математических задач, охватывающий арифметику, теорию чисел, алгебру, линейную алгебру, математический анализ и статистику, в соответствии со стандартом Common Core Standards. Доступны три уровня сложности: начальный, средний и продвинутый. В блоке генерации предлагается

[1] Wolfram Generator. URL: https://www.wolframalpha.com/pro/problem-generator/

задача, правильный ответ на которую нужно ввести в соответствующее поле. Даются три попытки, при ошибке генерируется новая задача. История ответов отображается в нижней части экрана. Отсутствуют задачи на вычисление пределов, невозможно задать класс функций для интегрирования или нахождения производных, не включены задачи из вычислительной математики. Для доступа к полному набору функций необходима платная подписка.

Algebramaker.com предоставляет возможность генерации задач по алгебре, исчислению, арифметике, тригонометрии. В задачах на пределы предлагаются задачи исключительно на отношение многочленов, имеется немногочисленное количество задач на вычисление производных и интегрирование от ограниченного класса функций. Задачи из области вычислительной математики не представлены.

Генерации задач посвящены диссертационные исследования и разработки. В исследовании Ю. А. Зорина [2] был разработан язык GILT для описания алгоритмов генерации заданий. Создание визуального генератора под названием «Платан» разрабатывалась с использованием языка JavaScript и применением библиотеки jQuery. И. А. Посов в своем диссертационном исследовании [3] создал язык Possum, который поддерживал систему компьютерной алгебры Maxima и в качестве базового языка программирования содержал JavaScript. Н. А. Иванова и Н. Н. Сосновский в своих работах [4; 5] для генерации материалов используют систему Mathematica, которая основывается на языках C, C++, Java и специальном языке для символьных вычислений – Wolfram Language. Разработка [6] написана на языке JavaScript и содержит собственные библиотеки для пошагового вывода полученных результатов и отображения множества решений на числовых осях для различных неравенств. В работе [7] авторы для создания задач на платформе GeoLin выбрали язык программирования Python. Выбор обосновывался тем, что данный язык является наиболее популярным, что упрощает чтение программного кода для большинства пользователей.

В каждой из рассмотренных работ использовался специализированный язык программирования. Автор статьи акцентирует внимание на том, что для разработки генераторов в области математического анализа и вычислительной математики требуется оптимальный язык программирования с точки зрения производительности и функциональности.

Многие современные языки программирования содержат математические библиотеки, которые позволяют решать различного рода задачи, начиная от задач математического анализа, вычислительной математики, линейной алгебры и заканчивая выполнением сложных вычислений в области квантовой механики.

Целью работы является проведение эксперимента с определенным набором задач над математическими библиотеками языков веб-программирования JavaScript, PHP и языка общего назначения Python, направленного на выбор библиотек с необходимой функциональностью, максимальной производительностью для реализации генераторов задач в области некоторых тем математического анализа и вычислительной математики.

Математические библиотеки языков программирования

Современные языки программирования все больше помогают в научных вычислениях подобно математическим пакетам Wolfram Mathematica, Matlab, Maple. В работе [8] авторы сравнивали библиотеку Math.js с другими библиотеками языков программирования JavaScript, Python и C++, такими как Sylvester, numeric.js, ndarray, NumPy и Octave. Сравнительный анализ времени выполнения матричных операций этими библиотеками показал, что Math.js работала медленнее всех, а вычисление определителя значительно отставало от других библиотек.

В исследовании [9] была предложена альтернатива Matlab, состоящая из трех пакетов Python: SciPy, NumPy и Matplotlib. Сравнительный анализ эффективности Matlab и Python продемонстрировал, что Python отличается более понятным и компактным кодом, не требует по-

купки дорогостоящей лицензии, обладает широким разнообразием библиотек, которые делают его мощным незаменимым инструментом для научных работ.

При создании веб-приложений с акцентом на математические задачи целесообразно использовать языки программирования, включающие библиотеки с обширным набором функций для конкретной области, способных существенно упростить процесс создания подобных приложений, которые должны поддерживать символьные вычисления и численные методы для приближенных решений.

Символьные вычисления, также известные как символьная математика или компьютерная алгебра, позволяют манипулировать с математическими объектами и предоставлять результат в аналитическом (символьном) виде, например, для символьного интегрирования, упрощения формул. В символьных вычислениях квадратный корень из числа 2 будет записан в виде $\sqrt{2}$, а не в приближенном формате 1.41421.

Численные вычисления используют методы приближенного решения математических задач, предоставляя решение с определенной степенью точности, в случае, когда точные методы громоздки. Во многом они могут быть достаточно эффективными для сложных вычислений.

Наиболее известными языками программирования, которые используются в веб-разработке, являются JavaScript, PHP, Python. Каждый из них является интерпретируемым и для них написаны математические библиотеки. В данной статье делается акцент на библиотеках, способных предоставлять перечень функций для решения некоторых задач из области математического анализа (вычисление пределов, производных функций, неопределенных и определенных интегралов) и вычислительной математики (решение нелинейных уравнений, численное интегрирование, интерполирование функций, решение СЛУ).

JavaScript встраивается в HTML и выполняется в браузере, что позволяет создавать интерактивные элементы веб-страниц. Для данного языка разработаны математические библиотеки: Math.js, Algebrite и Nerdamer.

Math.js [2] является универсальной математической библиотекой, которая обладает гибким анализатором выражений с поддержкой символьных вычислений, включает в себя обширный набор встроенных функций и констант; предлагает комплексное решение для работы с различными типами данных, такими как числа, большие числа, комплексные числа, дроби, единицы измерения, строки, массивы и матрицы; включает в свой состав следующие функции: алгебраические (нахождение производных, решение линейных уравнений), арифметические (базовые операции сложения, вычитания, умножения и деления и сложные функции, такие как возведение в степень, вычисление квадратного корня и другие), тригонометрические, битовые, комбинаторные, комплексные, матричные (операции с матрицами, такие как сложение, умножение, вычисление определителя и другие), статистические, а также функции вероятности.

Algebrite [3] – расширяемая библиотека JavaScript для символьных вычислений, содержит функции для работы с дробями, комплексными числами в прямоугольной и полярной формах. Библиотека позволяет упрощать математические выражения, находить символьные и численные корни полиномов, поддерживает работу с единицами измерения, матрицами и тензорами, производными и градиентами, определенными, неопределенными и кратными интегралами.

Nerdamer [4] – библиотека, поддерживающая символьные вычисления, предоставляет широкий спектр функций, выполняющих сложные математические операции. Содержит следующие особенности: выполнение символьных вычислений, таких как дифференцирование, интегрирование, разложение на множители; решение нелинейных, линейных уравнений и систем уравнений; поддерживает множество математических функций (тригонометрические, гипер-

[2] Math.js. URL: https://mathjs.org/

[3] Algebrite. URL: http://algebrite.org/

[4] Nerdamer. URL: https://nerdamer.com/

болические и другие); предоставляет функции для работы с матрицами, комплексными числами.

Язык PHP часто применяется для серверной веб-разработки. PHP-код встраивается в HTML и обрабатывается на стороне сервера, формируя HTML, который передается на клиентскую сторону[5]. PHP включает мощную математическую библиотеку MathPHP[6], которая включает в себя множество функций для работы с арифметическими выражениями, полиномами, матрицами и векторами, комплексными числами. Содержит функции для выполнения интерполяции, численного дифференцирования и интегрирования, поиска корней. MathPHP также предлагает полезные функции в области вероятности и статистики.

Python используется в различных задачах, включая веб-разработку, научные вычисления, анализ данных, машинное обучение и многое другое. Для языка Python существуют три известные библиотеки для выполнения научных расчетов: NumPy, SymPy, SciPy.

NumPy[7] является фундаментальным пакетом для научных вычислений на Python. Включает в себя набор функций для эффективных операций с массивами, включая математические и логические операции, манипуляции с формами, сортировку, выбор, ввод-вывод, дискретные преобразования Фурье, основные элементы линейной алгебры, статистические операции, генерацию случайных чисел и многое другое.

SymPy[8] – библиотека Python предоставляет возможности для символьных вычислений, позволяя осуществлять символьное дифференцирование, интегрирование, нахождение пределов, упрощение выражений, решение уравнений и многие другие. Имеет возможность форматировать результат вычислений в виде кода LaTeX.

SciPy[9] является библиотекой, которая дополняет возможности NumPy и содержит модули для оптимизации, численного интегрирования, интерполяции, дифференциальных уравнений и многих других.

Методология эксперимента

Для проведения эксперимента в области математического анализа и вычислительной математики было выбрано по три различные задачи по каждому типу задач по определенной теме. Задачи по математическому анализу [10, с. 1–5] содержали задачи из задачника Л. А. Кузнецова [11] на следующие темы:

- пределы числовых последовательностей (предел отношения двух многочленов – задача 2; предел от иррациональностей – задача 3, задача 4; второй замечательный предел – задача 6;
- пределы функций (предел отношения двух многочленов – задача 9; первый замечательный предел – задача 11; нахождение пределов с использованием замены переменной – задача 12; пределы от эквивалентных бесконечно малых – задача 14, задача 17; второй замечательный предел – задача 16;
- производные (производные от частного, разности, суммы, сложной функции – задача 5; производные от показательных, тригонометрических и степенных функций – задача 6; производные от логарифмов – задача 7; производные от тригонометрических функций – задача 8; производные от обратных тригонометрических функций и степенных функций – задача 9; производные от сложных функций, гиперболических функций, обратных тригонометрических функций – задача 10;

[5] Официальный сайт php. URL: https://www.php.net/manual/en/intro-whatis.php

[6] Репозиторий GitHub – mathPHP. URL: https://github.com/markrogoyski/math-php

[7] NumPy. URL: https://numpy.org

[8] SymPy. URL: https://www.sympy.org/en/index.html

[9] SciPy. URL: https://scipy.org

- интегралы (метод интегрирования по частям – задача 1, задача 2; интегрирование заменой переменной – задача 3, задача 4; интегрирование рациональных дробей – задача 5; универсальная тригонометрическая подстановка – задача 8; $\int \sin^m x \cdot \cos^n c\, dx$ – задача 10; интегрирование от иррациональных функций с помощью тригонометрической подстановки – задача 12.

Задачи по вычислительной математике [10, с. 6–7] содержали задачи из задачников [12; 13] на следующие темы: решение нелинейных уравнений (метод дихотомии (метод половинного деления), метод Ньютона, метод хорд) [12, с. 60], численное интегрирование (метод трапеции, метод Симпсона (метод парабол), метод Ньютона (правило 3/8)) [13, с. 93–94], интерполирование функций (интерполяционные полиномы Лагранжа и Ньютона) [13, с. 112–113], решение систем линейных уравнений (СЛУ) (метод Гаусса, метод LU-разложения) [10, с. 71].

При генерации обширного количества задач с ответами нужно учитывать время выполнения вычислений (производительность). Это позволит за минимальное время осуществить генерацию задач и не заставит пользователя находиться в длительном процессе ожидания.

Оценка производительности математических библиотек осуществлялась определением скорости предоставления результатов путем измерения времени выполняемого кода с использованием определенных конструкций для каждой отдельной задачи.

Функциональность языков программирования определялась из соответствующих функций математических библиотек, которые применялись для вычисления конкретных задач. Наглядное представление синтаксиса, используемого в каждой библиотеке для различных вычислительных задач, показано на рис. 1. Видно, что каждая отдельная библиотека не содержит всех необходимых функций для решения задач одновременно в символьном и численном представлениях.

Обновление библиотек является важным показателем того, что разработчики следят за созданием новых функций, улучшений, обеспечением совместимости с новыми версиями языков программирования, исправлением ошибок, которые увеличивают стабильность кода. Регулярность периодов обновления библиотек на май 2024 года представлена в табл. 1.

Таблица 1

Регулярность обновления библиотек

Table 1

Regularity of updating libraries

Математическая библиотека	**Регулярность**	**Последняя версия**
Math.js	регулярные обновления	v12.4.2
Algebrite	последнее обновление в 2021 г.	v1.4.0
Nerdamer	последнее обновление в 2022 г.	v1.1.13
MathPHP	регулярные обновления	v.2.10.0
NumPy	регулярные обновления	v.1.26.4
SymPy	регулярные обновления	v.1.12.1rc1
SciPy	регулярные обновления	v.1.13.0

Из анализа табл. 1 видно, что библиотеки Algebrite и Nerdamer уже несколько лет не обновлялись, из этого следует, что набор функций не расширялся и они могут не совсем эффективно решать конкретную задачу. Данный факт подчеркивает значимость текущего эксперимента.

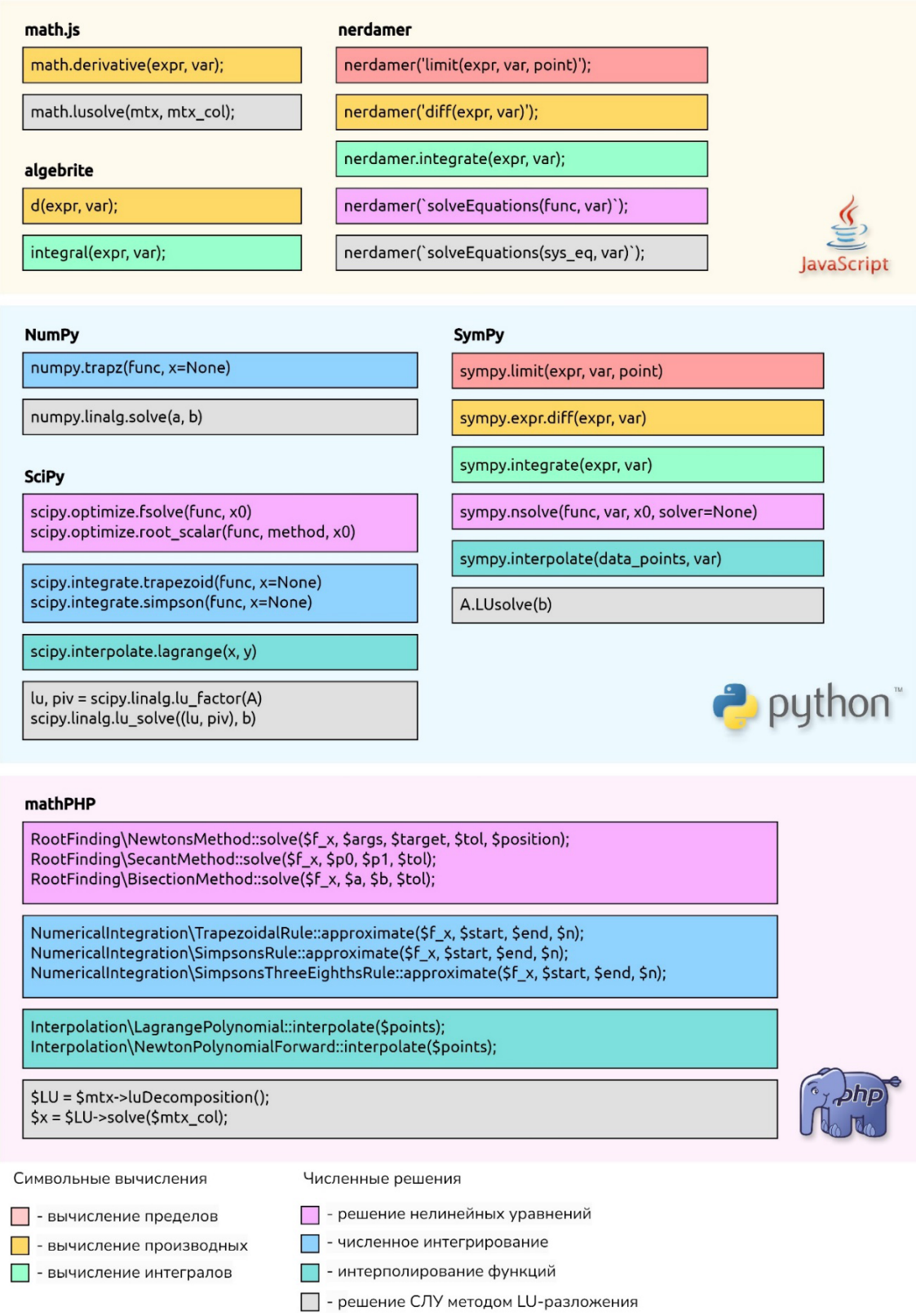

Рис. 1. Синтаксические конструкции для соответствующих задач на различных языках программирования
Fig. 1. Syntactic constructs for appropriate tasks in different programming languages

Результаты эксперимента

Проведение эксперимента было произведено на компьютере со следующими характеристиками:

- процессор (CPU): Intel Core i5-9600KF, 6 ядерный процессор с тактовой частотой 3,7 ГГц;
- оперативная память (RAM): две платы по 8 ГБ DDR4 2666 МГц;
- видеокарта (GPU): GeForce GTX 1650 SUPER с 4 ГБ GDDR6;
- жесткий диск: SSD на 240 Гб + HDD на 1 ТБ;
- операционная система: Microsoft Windows 10 Pro.

Измерение скорости выполнения измерялось в секундах до семи цифр после запятой. С подробной таблицей измерения производительности по каждой задаче для символьных и численных решений можно ознакомиться в репозитории Zenodo в таблице DATA.xlsx [10]. Результаты измерения производительности математических библиотек в задачах, требующих символьных вычислений с общим временем для каждой темы, представлены в табл. 2. Красным цветом отмечены результаты, в которых при вычислениях были допущены ошибки, а также в случаях, когда библиотеки не смогли справиться с одной из задач. Обозначение в виде X означает, что данный тип задач не поддерживается в библиотеке.

Таблица 2

Производительность математических библиотек
при выполнении символьных вычислений

Table 2

Mathematical libraries performance when performing symbolic calculations

Наименование задачи	**JavaScript**			**Python**
	Math.js	**Algebrite**	**Nerdamer**	**SymPy**
Пределы	X	X	0,7841000	2,4984908
Производные	0,6054000	4,7926000	2,6580000	0,2505858
Интегралы	X	0,8938000	0,1682000	3,4683402
Общее время	0,6054000	5,6864000	3,6103000	5,9668310

В библиотеке Math.js нет встроенных функций для вычисления пределов и интегралов. Существует расширение mathjs-simple-integral[10], которое позволяет вычислять интегралы. Этот пакет полностью поддерживает полиномиальные выражения, но все еще находится в бета-версии и обладает некоторыми существенными ограничениями, такими как отсутствие реализации интегрирования по частям или u-подстановки. Вычисление производных в Math.js имеет длинное решение, после упрощения ничего не изменяется (рис. 2), что сказывается на эффективности решения задач.

```
y'= (15 * (x + 1) ^ (-1 / 2) * (18 * x ^ 2 + 16 * x - 2) + (x + 1) ^ (-3 / 2) * (6 * x ^ 3 + 8 * x ^ 2 - 4 - 2 * x) * -15 / 2) / 225   math.js derivative.html:30
y'= (8 * (-(3 * x) - 2) * (x - 1) ^ (1 / 2) / x ^ 3 + 4 * (3 * (x - 1) ^ (1 / 2) + (3 * x + 2) * (x - 1) ^ (-1 / 2) / 2) / x ^ 2) / 16   math.js derivative.html:36
y'= (x ^ 2 + 2) ^ (-1 / 2) * (x ^ (-1 / 2) / 2 + 3) + (-(3 * x ^ 2) - x ^ (3 / 2)) * (x ^ 2 + 2) ^ (-3 / 2)   math.js derivative.html:42
```

Рис. 2. Math.js вычисление производных
Fig. 2. Math.js calculation of derivatives

[10] https://github.com/joelahoover/mathjs-simple-integral

Algebrite не поддерживает функции для расчета пределов. В некоторых задачах на производные Algebrite даже после упрощения получаются громоздкие решения (рис. 3). При вычислении интегралов с использованием Algebrite из 24 задач было корректно решено только 6 задач, наличие тригонометрических и показательных функций приводит к ошибкам.

```
y'= 1-exp(x)/((1+exp(x)+exp(2*x))^(1/2)*(2+exp(x)+2*   algebrite_derivative.html:45
(1+exp(x)+exp(2*x))^(1/2)))-exp(x)/(2+exp(x)+2*
(1+exp(x)+exp(2*x))^(1/2))-2*exp(2*x)/((1+exp(x)+exp(2*x))^(1/2)*(2+exp(x)+2*
(1+exp(x)+exp(2*x))^(1/2)))
```

Рис. 3. Algebrite вычисление производных
Fig. 3. Algebrite calculation of derivatives

Nerdamer в задачах на вычисление пределов допускал ошибки и с одной задачей не справился, всего из 30 задач успешно было решено только 11 задач. Nerdamer лучше вычисляет пределы от отношения многочленов. Пример неудачного решения предела показан на рис. 4.

```
Предел 10:  e^(-106*limit((-1+3*n^2+4*n)^(-1)*              nerdamer_limits.html:92
(2*n+3*n^2+7)^(-1)*n,n,Infinity)-116*limit((-1+3*n^2+4*n)^(-1)*
(2*n+3*n^2+7)^(-1)*n^3,n,Infinity)-152*limit((-1+3*n^2+4*n)^(-1)*
(2*n+3*n^2+7)^(-1)*n^2,n,Infinity)-18*limit((-1+3*n^2+4*n)^(-1)*
(2*n+3*n^2+7)^(-1)*n^5,n,Infinity)-18*limit((-1+3*n^2+4*n)^(-1)*
(2*n+3*n^2+7)^(-2)*n,n,Infinity)-60*limit((-1+3*n^2+4*n)^(-1)*
(2*n+3*n^2+7)^(-1)*n^4,n,Infinity)+214*limit((-1+3*n^2+4*n)^(-1)*
(2*n+3*n^2+7)^(-2)*n^2,n,Infinity)+234*limit((-1+3*n^2+4*n)^(-1)*
(2*n+3*n^2+7)^(-2)*n^6,n,Infinity)+498*limit((-1+3*n^2+4*n)^(-1)*
(2*n+3*n^2+7)^(-2)*n^5,n,Infinity)+54*limit((-1+3*n^2+4*n)^(-1)*
(2*n+3*n^2+7)^(-2)*n^7,n,Infinity)+618*limit((-1+3*n^2+4*n)^(-1)*
(2*n+3*n^2+7)^(-2)*n^3,n,Infinity)+718*limit((-1+3*n^2+4*n)^(-1)*
(2*n+3*n^2+7)^(-2)*n^4,n,Infinity))
```

Рис. 4. Nerdamer проблема при вычислении предела
Fig. 4. Nerdamer problem when calculating the limit

Решение производных в Nerdamer в некоторых случаях сопровождалось сложными решениями и некорректными записями (рис. 5). Большинство задач на вычисление производных от сложных функций влекут за собой ошибки. В задачах на интегрирование в Nerdamer из 24 задач успешно решено 10 задач. Ошибок не наблюдалось только в типах задач, представленных в задаче 3.

Библиотека SymPy языка программирования Python успешно справилась со всеми предложенными задачами по пределам, производным и интегрированию по сравнению с рассмотренными библиотеками.

```
y'= 3*(1+e^(2*x)+e^x)*(119696244^(-                    nerdamer_derivative.html:50
x)*325368125^x*sqrt(1+119696244^(-2*x)*325368125^(2*x)+119696244^(-
x)*325368125^x)+2+2*119696244^(-2*x)*325368125^(2*x)+2*119696244^(-
x)*325368125^x+2*sqrt(1+119696244^(-2*x)*325368125^(2*x)+119696244^(-
x)*325368125^x))*(2+2*sqrt(1+e^(2*x)+e^x)+e^x)
```

Рис. 5. Nerdamer решение производных
Fig. 5. Nerdamer solution of derivatives

Таблица 3

Производительность математических библиотек при выполнении численных вычислений

Table 3

Mathematical libraries performance in numerical computing

Наименование задачи		**JavaScript**		**PHP**	**Python**		
		Math.js	**Nerdamer**	**MathPHP**	**NumPy**	**SymPy**	**SciPy**
Решение нелинейных уравнений	Численный метод, выбранный алгоритмом библиотеки	X	0,0260000	X	X	0,0360805	0,0020579
	Метод Ньютона	X	X	0,0008490	X	X	0,0329740
	Метод хорд	X	X	0,0007200	X	X	0,0089337
	Метод половинного деления	X	X	0,0663150	X	0,0126936	0,0053973
Численное интегрирование	Метод трапеции	X	X	0,0076079	0,0002535	X	0,0004673
	Метод Симпсона (парабол)	X	X	0,0681238	X	X	0,0017882
	Метод Ньютона (правило 3/8)	X	X	0,0573912	X	X	X
Интерполирование функций	Интерполяционный полином Лагранжа	X	X	0,0019729	X	0,0000428	0,0000024
	Интерполяционный полином Ньютона	X	X	0,0021669	X	X	X
Решение СЛУ	Метод LU-разложения	0,0198000	0,0340000	0,0672503	0,0001408	0,0761548	0,0068897
Общее время		0,0198000	0,0600000	0,2723970	0,0003943	0,1249717	0,0585105

В ходе сравнительного анализа библиотек JavaScript и Python для символьных вычислений было обнаружено, что только библиотека SymPy эфективно справилась с задачами, связанными с пределами и интегралами. При рассмотрении задач на производные и сравнении библиотек Math.js, Algebrite и SymPy, было выявлено, что производительность SymPy значительно превосходит остальные. Важно отметить, что максимальное время выполнения программы в библиотеке Algebrite составляет около 6 секунд суммарно для каждого типа задач, содержащих по 3 различных варианта, причем задачи на пределы не учитываются. Вычисление задач на производные занимает почти 5 секунд, в то время как в SymPy на их вычисление уходит четверть секунды. Если в Algebrite потребуется создать n вариантов подобного рода задач, время выполнения значительно увеличится, что негативно скажется на производительности.

Производительность вычислений математическими библиотеками в задачах, требующих численных методов, представлены в табл. 3. Рассматривая тему, посвященную решению нелинейных уравнений одним из трех методов, заметно, что все методы поддерживают только библиотеки MathPHP и SciPy. Библиотека Algebrite не поддерживает решения нелинейных уравнений, она способна решать только полиномиальные уравнения высоких степеней.

Библиотека Nerdamer при решении нелинейных уравнений возвращает в ответе первый набор решений, удовлетворяющий ограничениям. Следует отметить, что при работе с этой библиотекой могут возникнуть некоторые ошибки с плавающей точкой. Во время экспериментального исследования при решении нелинейных уравнений было обнаружено, что вместо ожидаемого одного корня получались два приближенных корня, аналогично в случаях, когда предполагалось два корня, было выведено четыре.

В Nerdamer и SciPy существуют функции solveEquations и nsolve, в которых не указано, какой метод используется для вычисления нелинейного уравнения. В SymPy можно явно указать метод половинного деления, о других методах в документации не указано. Сравнительный анализ решений нелинейных уравнений численным методом, который выбирается алгоритмом конкретной библиотеки, показал значительную эффективность SciPy.

Оценивая скорость выполнения вычислений методом Ньютона, методом хорд в библиотеках MathPHP и SciPy можно убедиться в том, что в MathPHP производительность выше, чем в SciPy. Сравнивая MathPHP, SymPy и SciPy по решению задач методом половинного деления, видно, что быстрее всех справляется SciPy.

Численное интегрирование осуществлялось тремя методами. MathPHP поддерживает все методы, в отличие от остальных библиотек. Анализируя результаты по методу трапеции, в библиотеках MathPHP, NumPy и SciPy значительная производительность просматривается у NumPy. При сопоставлении MathPHP и SciPy по методу трапеции и методу Симпсона (парабол) можно заметить, что SciPy более продуктивна.

Осуществлять интерполирование функций с использованием полинома Лагранжа и полинома Ньютона способна только библиотека MathPHP. Интерполяционный полином Лагранжа поддерживают библиотеки MathPHP, SymPy и SciPy, из которых последняя во много раз эффективнее остальных.

Функции для решения систем линейных уравнений методом LU-разложения присутствуют во всех библиотеках, представленных в табл. 3, по скорости вычисления преобладает библиотека NumPy.

Заключение

Проведенный эксперимент показал, что не каждая математическая библиотека языков программирования JavaScript, PHP в полной мере поддерживает все функции для создания генераторов математических задач в рассмотренных темах. Python показал значительную эффективность, библиотеки NumPy, SymPy и SciPy делают его мощным как в символьных, так и в численных вычислениях.

Оптимальным вариантом выступает объединение возможностей Python и JavaScript, что дает возможность создавать динамичные веб-страницы, способные выполнять сложные вычислительные задачи. При помощи JavaScript можно обеспечивать интерактивность на стороне клиента, позволяя веб-страницам реагировать на действия пользователя в реальном времени, что входит в компетенции Frontend-разработчика. Использование языка Python на стороне сервера, курируемое Backend-разработчиком, предоставит способность проведения сложных расчетов. Данная комбинация языков программирования обеспечит возможность разработки эффективных и масштабируемых веб-приложений. Надеемся, что результаты эксперимента помогут многим разработчикам и исследователям при реализации генераторов математических задач.

Список литературы

1. **Винокурова Д. В.** Метод генерации уникальных вариантов для математических задач // Компьютерные инструменты в образовании. 2024. № 1. С. 71–84. DOI: 10.32603/2071-2340-2024-1-100
2. **Зорин Ю. А.** Автоматизация построения многовариантных тестовых заданий на основе деревьев И/ИЛИ: Дис. … канд. техн. наук. Томск, 2014. 139 с.
3. **Посов И. А.** Автоматизация процесса разработки и использования многовариантных учебных заданий: Дис. … канд. техн. наук. СПб., 2012. 134 с.
4. **Иванова Н. А.** Возможные направления применения ресурсов программирования среды Mathematica при решении математических задач // Вестн. Балт. федер. ун-та им. И. Канта. Серия: Филология, педагогика, психология. 2012. № 5. С. 155–160.
5. **Сосновский Н. Н.** Разработка методических материалов в среде системы Mathematica // Компьютерные инструменты в образовании. 2015. № 5. С. 53–60.
6. Свидетельство о государственной регистрации программы для ЭВМ № 2023618413 Российская Федерация. Программный модуль «Нахождение расположения корней квадратного уравнения с параметром». Версия 1.0 : No 2023616932 : заявл. 11.04.2023 : опубл. 25.04.2023 / Д. В. Винокурова.
7. **Гилев П. А., Казанков В. К., Табиева А. В.** Автоматическая генерация и проверка задач по дисциплинам математического цикла в высшей школе // Современное педагогическое образование. 2022. № 11. С. 142–147.
8. **De Jong J., Mansfield E.** Math.Js: An Advanced Mathematics Library For JavaScript // Computing in Science & Engineering. 2018. Vol. 20, no. 1. P. 20–32, 2018. DOI: 10.1109/mcse.2018.011111122
9. **Ranjani J., Sheela A., Meena K. P.** Combination of NumPy, SciPy and Matplotlib/Pylab – a good alternative methodology to MATLAB – A Comparative analysis // 1st International Conference on Innovations in Information and Communication Technology (ICIICT). 2019. DOI: 10.1109/ICIICT1.2019.8741475
10. **Винокурова Д. В.** Экспериментальное исследование производительности математических библиотек языков JavaScript, PHP и Python // Zenodo. 2024. DOI: 10.5281/zenodo.11402820
11. **Кузнецов Л. А.** Сборник заданий по высшей математике (типовые расчеты). М.: Высшая школа, 1994.
12. **Зенков А. В.** Вычислительная математика для IT-специальностей: Учеб. пособ. М.; Вологда: Инфра-Инженерия, 2022.
13. **Зализняк В. Е.** Теория и практика по вычислительной математике: Учеб. пособ. Красноярск: Сиб. федер. ун-т, 2012.

References

1. **Vinokurova D. V.** Method of Generating Unique Variants for Mathematical Problems. *Computer tools in education*, 2024, no. 1, pp. 71–84 (in Russ.); DOI: 10.32603/2071-2340-2024-1-100
2. **Zorin Y. A.** Automation of multivariate test case construction based on AND/OR trees. Cand. Sc. diss., TUSUR, Tomsk, 2014. (in Russ.)
3. **Posov I. A.** Automating the development and use of multivariate training tasks. Cand. Sc. diss., SPbSU, St. Petersburg, 2012. (in Russ.).
4. **Ivanova N. A.** Possible areas of application of Mathematica programming resources in solving mathematical problems. *Vestnik Baltijskogo federal'nogo universiteta im. I. Kanta. Seriya: Filologiya, pedagogika, psihologiya*, 2012, no. 5, pp. 155–160. (in Russ.)
5. **Sosnovskij N. N.** Development of methodical materialsin the environment of "mathematica". *Computer tools in education*, 2015, no. 5, pp. 53–60. (in Russ.)
6. **Vinokurova D. V.** Program module "Finding the location of roots of a quadratic equation with a parameter". Version 1.0. Apr. 25, 2023. (in Russ.)
7. **Gilev P. A., Kazankov V. K., Tabieva A. V.** Automatic generation and verification of problems on disciplines of mathematical cycle in higher school. *Modern Pedagogical Education*, 2022, no. 11, pp. 142–147. (in Russ.)
8. **De Jong J., Mansfield E.** Math.Js: An Advanced Mathematics Library For JavaScript. *Computing in Science & Engineering*, 2018, vol. 20, no. 1, pp. 20–32. DOI: 10.1109/mcse.2018.011111122
9. **Ranjani J., Sheela A., Meena K. P.** Combination of NumPy, SciPy and Matplotlib/Pylab – a good alternative methodology to MATLAB – A Comparative analysis. *1st International Conference on Innovations in Information and Communication Technology (ICIICT)*, 2019. DOI: 10.1109/ICIICT1.2019.8741475
10. **Vinokurova D. V.** Experimental study of the performance of mathematical libraries of JavaScript, PHP and Python languages. *Zenodo*, 2024. DOI: 10.5281/zenodo.11402820. (in Russ.)
11. **Kuznetsov L. A.** Collection of tasks in higher mathematics (typical calculations). Moscow, Vysshaya Shkola publ., 1994. (in Russ.)
12. **Zenkov A. V.** Computational mathematics for IT specialties: a textbook. Moscow; Vologda, Infra-Inzheneriya publ., 2022. (in Russ.)
13. **Zaliznyak V. E., Shchepanovskaya G. I.** Theory and practice in computational mathematics: a textbook. Krasnoyarsk, Siberian Federal University publ., 2012. (in Russ.)

Сведения об авторе

Винокурова Дарья Валентиновна, аспирант

Information about the Author

Darya Valentinovna Vinokurova, PhD Student

Статья поступила в редакцию 06.06.2024;
одобрена после рецензирования 18.10.2024; принята к публикации 18.10.2024

The article was submitted 06.06.2024;
approved after reviewing 18.10.2024; accepted for publication 18.10.2024

Научная статья

УДК 004.891
DOI 10.25205/1818-7900-2024-22-3-28-39

Дообучение модели CodeBERT для написания комментариев к SQL-запросам

Данила Александрович Комлев

НИТУ МИСИС, Москва, Россия

komlevdanila742@gmail.com

Аннотация

Автоматизированное создание комментариев к исходному коду – актуальная тема в разработке программного обеспечения, где модели машинного перевода применяются для «перевода» кода в текстовые описания. Предобученная на шести языках программирования модель CodeBERT используется для поиска кода, генерации документации, исправления ошибок. Эта модель хорошо понимает семантики естественного языка, языков программировани, а также связи между ними, эта модель хорошо подходит для дообучения на различные прикладные задачи, связанные с кодом. В статье рассматривается дообучение модели CodeBERT для генерации комментариев к SQL-запросам. Эта задача является актуальной, так как в крупных проектах может использоваться множество SQL-запросов различной сложности, и комментарии помогают улучшить их читаемость и понимание. Однако ручное написание и поддержание актуальности комментариев требует времени и усилий разработчиков. В статье предложено использовать предобученную модель CodeBERT для автоматической генерации комментариев к SQL-коду, что сократит время и позволит поддерживать комментарии в актуальном состоянии. Для дообучения используются открытые датасеты, содержание SQL-запрос, а также комментарий к нему. Результаты тестирования показали, что дообученная модель успешно справляется с задачей создания комментариев к SQL-запросам, что также подтверждается полученными значениями метрики Bleu.

Ключевые слова

SQL-запрос, CodeBERT, Transformers, NLP, Bleu, дообучение, генерация комментариев

Для цитирования

Комлев Д. А. Дообучение модели CodeBERT для написания комментариев к SQL-запросам // Вестник НГУ. Серия: Информационные технологии. 2024. Т. 22, № 3. С. 28–39. DOI 10.25205/1818-7900-2024-22-3-28-39

Further Training of the CodeBERT Model for Writing Comments on SQL Queries

Danila A. Komlev

NUST MISIS, Moscow, Russian Federation

komlevdanila742@gmail.com

Abstract

Automated creation of comments to the source code is an urgent topic in software development, where machine translation models are used to “translate” code into text descriptions. The CodeBERT model, pre-trained in six programming languages, is used to search for code, generate documentation, and correct errors. This model understands well the semantics of natural language, programming languages, as well as the connections between them, this model is well suited for additional training on various applied tasks related to code. The article discusses the further training of the

CodeBERT model for generating comments on SQL queries. This task is relevant, since large projects can use many SQL queries of varying complexity, and comments help to improve their readability and understanding. However, manually writing and keeping comments up-to-date takes time and effort from developers. The article suggests using the pre-trained CodeBERT model to automatically generate comments on SQL code, which will reduce time and allow you to keep comments up to date. For further training, open datasets, the contents of the SQL query, as well as comments on it are used. The test results showed that the pre-trained model successfully copes with the task of creating comments to an SQL query, which is also confirmed by the obtained values of the Bleu metric.

Keywords

SQL query, CodeBERT, Transformers, NLP, Bleu, additional training, comment generation

For citation

Komlev D. A. Further training of the CodeBERT model for writing comments on SQL queries. Vestnik NSU. Series: *Information Technologies*, 2024, vol. 22, no. 3, pp. 28–39 (in Russ.) DOI 10.25205/1818-7900-2024-22-3-28-39

Введение

В большинстве программных продуктов для персистентного хранения данных используются реляционные базы данных, например MS SQL, Postgresql, MySQL.

Для манипуляции данными в реляционных базах данных применяется SQL (Structured Query Language).

SQL-запросы бывают двух видов: DDL и DML:

1. DDL (Data Definition Language) используется для определения структуры и организации данных в базе данных. Это включает в себя создание, изменение и удаление объектов базы данных, таких как таблицы, индексы, представления, функции и т. д.

2. DML (Data Manipulation Language) используется для манипулирования данными внутри таблиц базы данных. Это включает в себя операции вставки (INSERT), обновления (UPDATE), удаления (DELETE) и выборки (SELECT) данных.

Например, для того чтобы получить Id, Username и Email для пользователей, зарегистрировавшихся, начиная с 2023-01-01, используется такой запрос:

```
SELECT UserID, Username, Email
FROM Users
WHERE RegistrationDate >= ‘2023-01-01’;
```

В больших программных продуктах может использоваться огромное число SQL-запросов разной сложности.

Для облегчения работы с этими запросами разработчики часто добавляют комментарии, чтобы улучшить читаемость и понимание кода. Однако написание комментариев к SQL-коду является рутинной задачей, которая может занимать значительное время разработчиков и аналитиков данных. Более того, часто возникают сложности с поддержанием актуальности комментариев: при изменении SQL-запроса комментарий к нему может остаться необновленным, что создает путаницу и затрудняет понимание кода.

В свете этих проблем актуальной задачей становится автоматизация процесса написания комментариев к SQL-коду с применением нейронных сетей. Это не только поможет сэкономить рабочее время программистов, но и обеспечит актуальность и консистентность комментариев, что приведет к улучшению процесса разработки и поддержки программного обеспечения.

Обзор модели CodeBERT

В данной работе будет использована предобученная модель CodeBERT. Эта модель разработана исследовательской командой в Microsoft. Модель специализируется на NLP-задачах, связанных с программным кодом:

- Code-To-Code – дополнение кода или перевод с одного языка программирования на другой;
- Code-To-Text – написание комментариев к коду;
- Text-To-Code – поиск кода;
- Text-To-Text − машинный перевод, генерация текста.

CodeBERT обучена на шести языках программирования: Python, Java, JavaScript, PHP, Ruby, Go и хорошо понимает как семантику программного кода, так и семантику человеческого языка. Суммарный объем данных, на которых обучалась модель, – более миллиона записей.

На языке SQL модель CodeBERT не обучалась. Также отличительной особенностью языка SQL является то, что SQL – не язык программирования, а язык запросов. А также, все языки, на которых обучалась модель, являются императивными, т. е. содержат прямые указания, что должна делать программа, а язык SQL является декларативным [1], т. е. с помощью языка описывается ожидаемый результат, а не способ его получения. Другими примерами декларативных языков являются HTML и XML.

Рассмотрим архитектуру CodeBERT. BERT (Bidirectional Encoder Representations from Transformers) – это модель глубокого обучения, разработанная на основе трансформерной архитектуры, которая способна эффективно анализировать контекст текстовых данных. Ее особенность заключается в том, что она обучается на больших объемах текста с использованием двунаправленных кодировщиков, что позволяет модели учитывать контекст как слева, так и справа от текущего слова или фразы. Это позволяет BERT создавать глубокие контекстуальные представления слов и фраз, что делает ее одной из наиболее эффективных моделей для широкого спектра задач обработки естественного языка.

Механизм внимания (attention) является ключевой особенностью архитектуры Transformers. При каждом новом предсказании трансформер оценивает контекст, фокусируя внимание на тех словах, которые являются наиболее значимыми для текущего шага. Эта оценка осуществляется путем вычисления весов внимания для каждого слова, отражающих его относительную важность.

Внимание позволяет модели определять вес для каждой позиции, сопоставляя запросы с ключами всех слов во входной последовательности. После нормализации весов с помощью функции softmax значения взвешиваются для создания контекстуально релевантных представлений.

Этот механизм заменил традиционные рекуррентные нейронные сети (RNN) благодаря своей способности параллельно обрабатывать длинные последовательности. RNN из-за рекурсивной природы часто испытывают трудности с захватом долгосрочных зависимостей. В отличие от них трансформеры, благодаря механизму внимания, эффективно анализируют и ближайшие, и отдаленные связи.

По сравнению с RNN трансформеры обрабатывают информацию быстрее, так как позволяют каждой позиции в последовательности напрямую взаимодействовать с любой другой позицией. Это решает проблему обучения в параллельных вычислениях, что особенно критично при работе с большими наборами данных. Трансформеры способны учитывать сложные взаимосвязи между словами независимо от их относительной позиции, что позволяет им успешно выполнять такие задачи, как машинный перевод, генерация текста и многие другие задачи обработки естественного языка.

У моделей BERT есть два этапа разработки:

- предобучение (pre-training) – обучение модели на большом объеме неразмеченных данных, которое помогает модели понять языковые закономерности и создать общее представление о языке;
- дообучение (fine-tuning) – донастройка модели для решения конкретной задачи. Для дообучения требуется значительно меньше данных, чем для предобучения.

В данной работе рассмотрен процесс дообучения предобученной модели CodeBERT для решения задачи написания комментариев к SQL-коду. Модель CodeBERT понимает семантики как языков программирования, так и семантики естественного языка, а также их взаимосвязи. Модель ничего не знает про SQL, так как в процессе предоучения SQL не использовался. Однако общие знания о семантике языков программирования позволят модели с помощью относительно небольшого датасета научиться писать комментарии к SQL.

CodeBERT, подобно BERT, использует архитектуру Transformer [2]. Однако в отличие от BERT, который обучается на текстах естественного языка, CodeBERT обучается на текстах программного кода и сопутствующих текстовых описаниях. Это позволяет модели эффективно работать с бимодальными данными и улавливать связи между кодом и его описанием.

Архитектура CodeBERT включает в себя два важных метода обучения: Masked Language Modeling (MLM) [3] и ReplacedToken Detection (RTD):

1) Masked Language Modeling. MLM – это техника обучения языковых моделей, применяемая в NLP. Идея заключается в том, что часть токенов во входных данных случайным образом маскируется, а затем модель обучается прогнозировать эти замаскированные токены на основании контекста, это используется в процессе предобучения модели. MLM является обучением без учителя (self-supervised learning), так как никакой предварительной разметки данных не требуется. Наглядный пример того, как работает MLM, представлен на рис. 1;

2) Replaced Token Detection. RTD [4], так же как и MLM, является техникой, используемой в процессе предобучения языковой модели. Идея заключается в том, что во входной последовательности токенов часть токенов заменяется. Но не на [MASK], как в MLM, а на другой токен, относительно близкий по смыслу, но неверный. Например, токен «окно» может быть заменен на токен «стекло». Между этими токенами есть связь, но они означают разное. В процессе обучения модель должна научиться определять токены, которые были заменены, это улучшает понимание моделью контекста, а также семантики предложения.

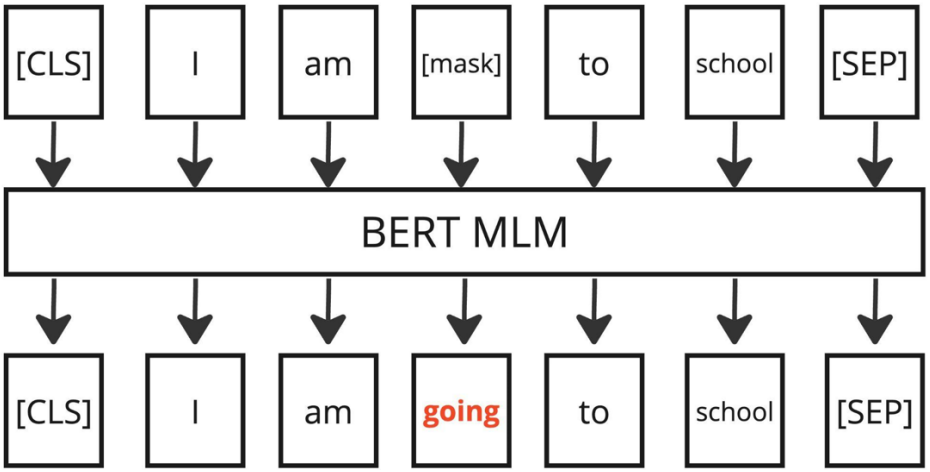

Рис. 1. Пример работы MLM
Fig. 1. An example of MLM operation

Токены замены генерируются, как правило, небольшой MLM-моделью, которая выдает близкий к оригинальному по смыслу токен. Пример работы RTD можно увидеть на рис. 2. Генератор с помощью MLM заменяет [MASK] на токены, близкие или равные исходным, а дискриминатор определяет, был ли исходный токен заменен.

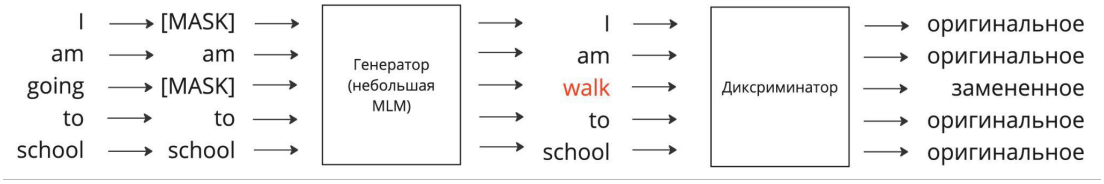

Рис. 2. Пример работы RTD
Fig. 2. An example of RTD operation

Как было отмечено ранее, CodeBERT является бимодальной моделью, так как работает с естественным языком и программным кодом. В связи с этим в RTD у CodeBERT используется не один генератор, как в базовой модели BERT, а два: для естественного языка и программного кода.

Оценка BLEU (BiLingual Evaluation Understudy) является метрикой в машинном переводе, используемой для оценки его качества [5]. Она представляет собой число в диапазоне от 0 до 1, где 0 соответствует низкому качеству перевода, а 1 – высокому.

Алгоритм BLEU основан на сравнении n-грамм [6] (обычно от 1 до 4) в сгенерированном тексте с эталонным текстом. Например, для фразы «я сегодня купил хлеб» и $n = 2$, имеем три биграммы:

1. «я сегодня»
2. «сегодня купил»
3. «купил хлеб»

Затем вычисляется пропорция n-грамм эталонного текста, которые присутствуют в сгенерированном тексте. Для расчета BLEU могут использоваться несколько наборов n-грамм, и итоговая оценка формируется как их взвешенная сумма.

BLEU учитывает не только совпадение слов, но и их количество и порядок. Это позволяет более точно оценивать качество перевода, учитывая контекст и семантику предложений.

Также BLEU штрафует короткие предложения по формуле 1:

$$\text{Brevity Penalty} = \begin{cases} 1, & if\ c > r \\ e^{(1-\frac{r}{c})}, & if\ c \le r \end{cases}. \quad (1)$$

Это нужно для того, чтобы короткие предложения, не имеющие смысла, например, состоящие из одного слова, не могли получить высокую оценку.

BLEU может быть представлена формулой (2):

$$\text{BLEU} = \text{BP} * \exp\left(\sum_{1}^{N} w_n * \log\left(p_n\right)\right), \quad (2)$$

где p_n – доля n-грамм эталонного текста, которые присутствуют в сгенерированном, а w_n – вес каждой n-граммы, как правило, $\frac{1}{N}$.

Для наглядности рассмотрим расчет метрики на примере:

Сгенерированное предложение: *A black cat is sleeping on the sofa*

Эталонное предложение: *The black cat is sleeping on the couch*

Рассчитаем n-граммы:

$$P_1 = \frac{6}{8} = \frac{3}{4}$$

$$P_2 = \frac{5}{7}$$

$$P_3 = \frac{4}{6} = \frac{2}{3}$$

$$P_4 = \frac{3}{5}$$

$$\text{BLEU} = \exp(\ln\left(\frac{3}{4}\right) + \frac{1}{2} * \ln\left(\frac{5}{7}\right) + \frac{1}{3} * \ln\left(\frac{2}{3}\right) + \frac{1}{4} * \ln\left(\frac{3}{5}\right) \approx 0,49 .$$

Однако стоит заметить, что в разных реализациях могут использоваться разные основания логарифма.

Предлагаемое решение

Для дообучения модели необходим датасет, содержащий в себе 2 колонки:

- SQL-запрос;
- комментарий к SQL-запросу на естественном языке.

Было найдено два открытых датасета:

1) https://huggingface.co/datasets/b-mc2/sql-create-context – датасет, содержащий сопоставление SQL-кода и человекопонятного описания этого кода;
2) https://huggingface.co/datasets/iamtarun/code_instructions_120k_alpaca – датасет сопоставления программного кода и описание этого кода на естественном языке. Код представлен на разных языках, в том числе на SQL.

Оба датасета хорошо подходят под нашу задачу, однако также в них есть некоторые особенности, которые негативно повлияют на дообучение модели, а именно:

1. Разные комментарии написаны в разном формате. Для близких по смыслу SQL-запросов могут быть совсем разные комментарии.

2. Встречаются строки, в которых написан не только SQL-запрос, но и код на каком-либо языке программирования. Как правило, это строки, в которых SQL-запрос являются частью программного кода.

3. В данных упоминаются разные диалекты SQL, например, MS SQL, Postgresql, MySql. В данной задаче нам не важны различия диалектов.

4. Встречаются строки в вопросной форме.

Для решения этих проблем была проведена предобработка данных, в рамках которой были выполнены следующие действия:

- были удалены все записи, в столбце комментария которых упоминаются языки программирования, фреймворки, а также другие, специфичные для программирования, термины, например: *api, endpoint, class*;
- были удалены все записи с комментарием в вопросной форме. Это те записи, которые заканчиваются ? и содержат одно из следующих слов: *what, where, which, who, how many, when;*
- комментарии были приведены к единой форме. Например, фразы: *dispay the, compute the, identify the, get the* были заменены на *find the*, так как *find the* чаще всего встречается в комментариях к запросам типа SELECT;
- из обоих столбцов были удалены управляющие последовательности, такие как \n и \t, обозначающие перенос строки и табуляцию;
- были удалены записи, содержащие слишком короткий или длинный комментарий или SQL-запрос;

- также весь текст был переведен в нижний регистр, так как SQL-запросы являются регистронезависимымы (кроме строковых литералов).

На рис. 3 представлены несколько записей из итогового датасета.

В финальном датасете 17323 записи. Из них по 1500 отводится на тестовую выборку и валидационную. Остальные записи (15323) отводятся на обучение модели.

В первом датасете присутствует колонка `context`, содержащая DDL-выражения, используемые для создания таблиц, которые фигурируют в SQL-запросах (колонка `answer`). Эта информация может быть полезна для генерации комментариев, однако текущая архитектура модели не предусматривает ее использования, модель принимает на вход только sql-запрос, для которого необходимо сгенерировать комментарий. Также во втором, более крупном датасете, информация о DDL-выражениях отсутствует.

С помощью датасета, описанного выше, будем дообучать модель.

Результаты

Для дообучения будет использоваться репозиторий CodeXGLUE [7], созданный Microsoft. Репозиторий предоставляет удобный интерфейс для взаимодействия с предобученной моделью CodeBERT, а также возможность дообучать модель под разные типы задач, в нашем случае «Code To Text» (генерация текстовых описаний из кода).

Модель Seq2Seq. Для этой цели мы используем архитектуру Seq2Seq, которая объединяет предобученный энкодер CodeBERT и новый декодер Transformer:

1. Энкодер (CodeBERT). Энкодер из CodeBERT служит основой, поскольку он уже предобучен на большом наборе данных и обладает глубокой способностью распознавать паттерны в коде. Этот этап подготовки позволяет эффективно извлекать ключевые признаки и семантическую информацию из входного кода. В процессе дообучения энкодер фокусируется на адаптации к более специализированным аспектам, необходимым для задачи «Code to Text».

	question	answer
0	find the most expensive product from the table...	select * from products order by price desc lim...
1	find all records from a table ordered by date ...	select * from table order by date asc;
3	create a stored procedure in insert new data i...	create procedure insert_user(in first_name var...
6	find the highest price for each product type.	select type, max(price) as 'highest_price'from...
8	sql query to find the top 5 highest vote counts	select name, vote from votes order by vote des...
...	...	...
20788	find the date for opponent in the final being ...	select date from table_name_5 where opponent_i...
20789	find the tournament for grass surface and oppo...	select tournament from table_name_11 where sur...
20790	find the fcc info on the radio frequency mhz 1...	select fcc_info from table_name_64 where frequ...
20791	find the mhz frequency of allapattah, florida.	select frequency_mhz from table_name_28 where ...
20792	find the attendance with date of june 11	select attendance from table_name_92 where dat...

Рис. 3. Датасет
Fig. 3. Dataset

2. Декодер. Для генерации текстовых описаний из кодовых фрагментов создается новый декодер на основе Transformer. Декодер содержит несколько слоев, чтобы эффективно обрабатывать сложные зависимости в последовательностях. В отличие от энкодера декодер начинается с нуля и будет обучен на специализированном датасете для конкретной задачи. В ходе тренировки он научится создавать детальные текстовые описания на основе выходных данных энкодера.

В качестве функции потерь используется кросс-энтропия [8], она измеряет расхождения между предсказанными вероятностями модели и истинными значениями. Значение кросс-энтропии считается по формуле:

$$H(p,q) = -\sum_{x \epsilon X} p(x) * \log\left(q(x)\right), \tag{3}$$

где $p(x)$ – истинная вероятность значения x, $q(x)$ – предсказанная вероятность значения x (0 или 1).

Функция потерь учитывает разницу между этими двумя вероятностями, где большие расхождения приводят к более высоким значениям кросс-энтропии. Если модель делает точные предсказания, вероятность $q(x)$ будет близка к истинной вероятности $p(x)$, что приводит к низкому значению кросс-энтропии. Таким образом, чем более уверена модель в правильном предсказании, тем меньше значение кросс-энтропии.

В контексте задачи данной статьи значение кросс-энтропии зависит от того, насколько точно модель предсказывает следующий токен в последовательности. Поскольку модель Seq2Seq работает с последовательными данными, каждый токен, предсказанный декодером, сравнивается с истинным следующим токеном в предложении.

В процессе генерации комментариев используется эвристический алгоритм Beam Search [9]. Этот алгоритм помогает эффективно формировать последовательности токенов для комментариев, выбирая на каждом шаге некоторое, заранее заданное количество наиболее вероятных токенов. Beam Search комбинирует элементы поиска в ширину и поиска в глубину, обеспечивая баланс между качеством и эффективностью предсказаний.

Для работы алгоритма задается параметр «ширина луча» (beam width), который определяет количество узлов, учитываемых на каждом шаге. При небольшом значении ширины луча алгоритм напоминает жадный поиск, быстро находя оптимальное решение, но с возможными компромиссами в точности и осмысленности генерируемых последовательностей. Увеличение ширины луча позволяет рассматривать большее число узлов, что повышает вероятность получения более релевантных токенов, но также увеличивает вычислительную нагрузку и время выполнения алгоритма.

Beam Search выполняет следующие основные шаги.

1. Инициализация: начинается с начального состояния, представляющего собой начальный токен (например, Start Of Sequence, SOS).

2. Расширение узлов: на каждом шаге алгоритм генерирует все возможные продолжения текущих последовательностей токенов.

3. Оценка вероятностей: каждое продолжение оценивается по вероятности с учетом предыдущих токенов и состояния модели.

4. Отбор лучших гипотез: из всех возможных продолжений отбирается фиксированное количество наиболее вероятных (beam width).

5. Проверка завершения: процесс повторяется, пока не достигнут конечный токен (End Of Sequence, EOS) или максимальная длина последовательности.

Пример работы алгоритма изображен на рис. 4

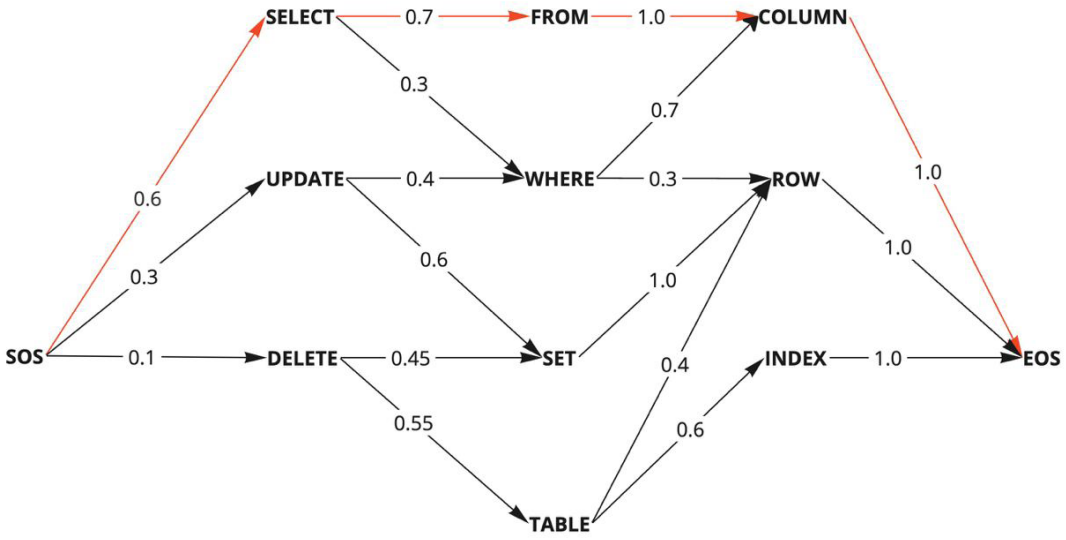

Рис. 4. Алгоритм Beam Search
Fig. 4. Beam Search Algorithm

Процесс дообучения, использующий модель Seq2Seq [10]:

- инициализируем предобученный энкодер и новый декодер с конфигурацией, включающей параметры для Beam Search;
- подготавливаем датасет, разделяя его на тренировочный и валидационный наборы, и настраиваем параметры оптимизатора и планировщика обучения;
- при тренировке модель обрабатывает каждый батч данных, рассчитывает потери и оптимизирует свои параметры посредством обратного распространения;
- модель дообучается на семи эпохах;
- после каждой эпохи производятся оценка и валидация, чтобы убедиться, что модель улучшает свое представление в задаче генерации текста из кода.

Выводы

После проведения дообучения модели CodeBERT для генерации комментариев к SQL-коду модель была проверена на тестовом наборе данных из 1000 записей.

Среднее значение метрики BLEU составило 0.45. Для понимания этого результата, отметим, что при значении BLEU выше 0.6 сгенерированный текст практически неотличим от текста, написанного человеком. Хотя наша модель еще не достигла этого уровня, результат 0.45 указывает на значительное улучшение качества генерации комментариев по сравнению с изначальным состоянием модели.

В таблице представлены некоторые результаты запуска модели на тестовом датасете. Из результатов видно, что модель выдает значения, хорошо описывающие SQL запрос. Даже в тех случаях, когда значение BLEU оценивается как низкое, сгенерированные комментарии все же предоставляют общее представление о содержании запроса, в некоторых случаях модель описывает sql-запрос не хуже, чем исходный комментарий, но другими словами.

Результаты генерации

Generation results

SQL-запрос	Исходный комментарий	Сгенерированный комментарий	BLEU
1	2	3	4
select count(loss) from table_name_87 where avg_g > 129.2	how much loss has an avg/g larger than 129.2	how much loss has a avg larger than 129.2	0.43
select sum(average) from table_name_99 where interview > 9.57 and swimsuit > 9.65	find the sum of average for interview more than 9.57 and swimsuit more than 9.65	find the sum of average for interview more than 9.57 and swimsuit more than 9.65	1.0
create procedure getproduct(@num1 int, @ num2 int) as begin declare @result int select @result = @num1 * @num2 return @result end	create a sql stored procedure that accepts two integer numbers and returns the product of these two numbers.	create a sql stored procedure that returns the number of items from a given table.	0.30
-- create the users tablecreate table users (id serial primary key, name varchar(255) not null);-- create the posts tablecreate table posts (id serial primary key, user_id integer references users(id), content varchar(255) not null);	create a sql database with two tables, users and posts. the users table should contain fields for id and name, while posts should contain fields for user_id and content.	create a sql database that stores information about users and posts.	0.06
select pole_position from table_1140085_2 where rnd = 3	find the pole position for the rnd 3	find the pole position with rnd being 3	0.38
select top 5 salaryfrom table order by salary desc;	select the top 5 highest salaries from a table.	find the top 5 highest salaries from a table.	0.86
update users set age = 25 where age > 25;	create a update the age field of the users whose age is greater than 25.	update the age of all users older than 25.	0.00
select count(average) from table_name_21 where wickets > 537 and career = «1899/00-1925/26» and matches < 211	find the total number of average for wickets more than 537, career of 1899/00-1925/26 and matches less than 211	find the total number of average for wickets more than 537 and career of 1819/19/26	0.52
select t1.fname from authors as t1 join authorship as t2 on t1.authid = t2.authid join papers as t3 on t2.paperid = t3.paperid where t3.title like «%functional%»	find the first names of all the authors who have written a paper with title containing the word «functional».	Find the first names of all the authors who paper contains «function»	0.44

Окончание табл.

1	2	3	4
select title, count(rating) as rating_count from moviesgroup by title order by rating_count desclimit 3;	find the top 3 most popular movies from a database of movies with their ratings.	find the top 3 movies with the highest rating.	0.18
select t3.headquartered_city from store as t1 join store_ district as t2 on t1.store_id = t2.store_id join district as t3 on t2.district_id = t3.district_id group by t3.headquartered_city order by count(*) desc limit 1	find the city with the most number of stores.	find the city with the most count of stores	0.66

Следует также отметить, что, несмотря на меньшее количество данных для SQL по сравнению с другими языками программирования (в среднем на 1 порядок), на которых изначально обучалась модель, сгенерированные комментарии оказались осмысленными и достаточно точно описывающими SQL-код. Это подчеркивает способность модели адаптироваться к новым языковым задачам и генерировать релевантные комментарии даже при ограниченном объеме обучающих данных.

Таким образом, дообучение CodeBERT для задачи генерации комментариев к SQL-коду продемонстрировало значимые результаты. Текущая дообученная модель способна генерировать комментарии к SQL-запросам относительно небольшого размера. Однако с использованием более полного и качественного датасета реализованный процесс дообучения может показать более высокие значения метрики BLEU.

Также это подтверждает потенциал модели CodeBERT для дообучения на различные прикладные задачи, связанные с программным кодом.

Список литературы / References

1. What Is Declarative Programming? URL: https://codefresh.io/learn/infrastructure-as-code/declarative-vs-imperative-programming-4-key-differences/
2. **Vaswani O., Shazeer N., Parmar N. etc.** Attention Is All You Need. URL: https://arxiv.org/abs/1706.03762
3. Masked Language Modeling in BERT. URL: https://www.scaler.com/topics/nlp/masked-language-model-explained/
4. **Clark K., Luong M.-T. D.** Manning C Electra: pre-training text encoders as discriminators rather than generators. URL: https://openreview.net/pdf?id=r1xMH1BtvB
5. Natural Language Processing: Bleu Score. URL: https://www.baeldung.com/cs/nlp-bleu-score
6. N-граммы. URL: https://deepai.org/machine-learning-glossary-and-terms/n-gram
7. CodeXGLUE. URL: https://microsoft.github.io/CodeXGLUE/
8. Cross Enptropy. URL: https://ml-cheatsheet.readthedocs.io/en/latest/loss_functions.html
9. Beam Search. URL: https://d2l.ai/chapter_recurrent-modern/beam-search.html
10. **Zhu Qingfu, Zhang Weinan, Zhou Lianqiang, Liu Ting.** Learning to Start for Sequence to Sequence Architecture. 2016. URL: https://www.researchgate.net/publication/306357583_Learning_to_Start_for_Sequence_to_Sequence_Architecture

Сведения об авторе

Комлев Данила Александрович, студент магистратуры

Information about the Author

Danila A. Komlev, Graduate Student

Статья поступила в редакцию 23.05.2024;
одобрена после рецензирования 15.10.2024; принята к публикации 15.10.2024

The article was submitted 23.05.2024;
approved after reviewing 15.10.2024; accepted for publication 15.10.2024

Научная статья

УДК 550.832.9
DOI 10.25205/1818-7900-2024-22-3-40-48

Программно-аппаратные решения потоковой обработки данных для компенсации температурных дрейфов скважинного инклинометра «Луч»

Василий Сергеевич Литвинов[1]
Александр Александрович Власов[1,2,3]
Дмитрий Владимирович Тейтельбаум[1]

[1]Научно-производственное предприятие геофизической аппаратуры «Луч»

[2]Новосибирский государственный университет

[3]Институт автоматики и электрометрии СО РАН
Новосибирск, Россия

litvinov@looch.ru
a.vlasov@nsu.ru
teytelbaum@looch.ru

Аннотация

Новосибирское научно-производственное предприятие геофизической аппаратуры «Луч» разрабатывает и производит телеметрические системы, эксплуатируемые в процессе бурения нефтегазовых скважин. В их состав входит датчик ориентации (инклинометр), оценивающий положение прибора в скважине на основе сигналов с трех акселерометров и трех магнитометров. Модули системы работают в условиях повышенных температур (до 150 °C), и для обеспечения заданной погрешности измерений углов ориентации требуется компенсация температурных дрейфов. В данной работе приводятся оценки допустимых дрейфов показаний датчиков, температурная полиномиальная модель акселерометров и магнитометров, методика компенсации. Проведен эксперимент, по результатам которого определена пригодность используемой модели и методики.

Ключевые слова

инклинометр, компенсация, температурный дрейф, калибровка

Для цитирования

Литвинов В. С., Власов А. А., Тейтельбаум Д. В. Программно-аппаратные решения потоковой обработки данных для компенсации температурных дрейфов скважинного инклинометра «Луч» // Вестник НГУ. Серия: Информационные технологии. 2024. Т. 22, № 3. С. 40–48. DOI 10.25205/1818-7900-2024-22-3-40-48

Software and Hardware Solution for Stream Processing of Data for Compensation of Temperature Drifts of LWD Orientation Sensor «Looch»

Vasily S. Litvinov [1], Alexander A. Vlasov [1,2,3], Dmitry V. Teytelbaum [1]

[1] Scientific-production enterprise of geophysical equipment "Looch"

[2] Novosibirsk State University

[3] Institute of Automation and Electrometry SB RAS
Novosibirsk, Russian Federation

litvinov@looch.ru
a.vlasov@nsu.ru
teytelbaum@looch.ru

Abstract

The Scientific-production enterprise of geophysical equipment "Looch" develops and manufactures LWD telemetry systems used in the process of drilling oil and gas wells. They include an orientation sensor (inclinometer) that estimates the position of the device in the well based on signals from three accelerometers and three magnetometers. System modules operate at high temperatures (up to 120 °C), and temperature drift compensation is required to ensure the specified orientation measurement error. This paper provides estimates of allowable drifts of sensor readings, a temperature polynomial model of accelerometers and magnetometers, and a compensation technique. An experiment was carried out, the results of which determined the suitability of the model and methodology used.

Keywords

LWD orientation sensor, temperature drift, compensation, calibration

For citation

Litvinov V. S., Vlasov A. A., Teytelbaum D. V. Software and hardware solution for stream processing of data for compensation of temperature drifts of LWD orientation sensor «Looch». *Vestnik NSU. Series: Information Technologies*, 2024, vol. 22, no. 3, pp. 40–48 (in Russ.) DOI 10.25205/1818-7900-2024-22-3-40-48

Введение

В современном мире нефтегазовая промышленность имеет огромное значение для экономики и жизни людей. Развитие этой отрасли требует постоянного совершенствования технологий и методов добычи полезных ископаемых. Во время бурения скважин важное место занимает контроль их траектории для проведения по продуктивной части пласта-коллектора. В настоящее время развито применение телеметрических систем, проводящих измерения внизу колонны непосредственно в процессе бурения и передающих актуальные значения параметров на поверхность.

Инклинометрия является прямым способом контроля за траекторией проводящейся скважины и позволяет определить три угла ориентации бурового инструмента [1]. Зенитный угол определяет наклон относительно линии отвеса, азимут определяет направление в горизонтальной плоскости, угол установки отклонителя определяет поворот прибора вокруг собственной оси и используется для задания направления бурения, в котором происходит искривление траектории ствола скважины.

Новосибирское научно-производственное предприятие геофизической аппаратуры «Луч» производит собственные телеметрические системы, в состав которых входит узел инклинометра. В процессе бурения скважин измерительное оборудование может нагреваться до 150 °C, поэтому важно контролировать работоспособность приборов и качество измерений во всем диапазоне температур.

Например, для акселерометра JAE в документации приведены следующие верхние границы для дрейфов:

- масштабный коэффициент: ±300 ppm/°C при температурах от 100 °C;
- смещение нуля: ±100 мк g/°C.

При использовании таких датчиков при повышении температуры с комнатных 20 °C до 150 °C возможен дрейф с 1,00 g до 1,04 g в вертикальном положении датчика и с 0,00 g до 0,01 g в горизонтальном положении датчика. Такие отклонения критично сказываются на качестве измерений, поэтому требуют устранения.

В данной работе решается задача программной компенсации температурных дрейфов сигналов для инклинометра «Луч» с целью улучшения его потребительских качеств.

Для достижения результата нужно было пройти ряд этапов: изучение характеристик датчиков, разработка методики температурной калибровки, разработка программы для микроконтроллера; разработка ПО для осуществления температурной калибровки при производстве инклинометров.

Общие сведения об инклинометре «Луч»

В состав инклинометра НПП ГА «Луч» (рис. 1) входят три одноосевых датчика ускорения маятникового типа и один трехосевой феррозондовый магнитометр, размещенные в разных концах прибора.

Рис. 1. Инклинометр производства НПП ГА «Луч» и система координат прибора
Fig. 1. Inclinometer produced by NPP GA Luch and the coordinate system of the device

Это сделано для исключения влияния магнитного поля, создаваемого током через катушки акселерометров, на показания магнитометра. На шасси установлено два термодатчика, измеряющих температуру в удаленных друг от друга точках прибора. Также в состав прибора входят платы источников питания, аналого-цифрового преобразования. Обработка сигналов производится на микроконтроллере STM32L496. Интерфейс асинхронный, протокол взаимодействия проприетарный, унифицированный для всех компонент телеметрической системы.

Пользователи подключаются к прибору при помощи ПО Colibri5 [5], позволяющим производить опрос датчиков, калибровку, сервисное обслуживание.

Система координат прибора привязана к его корпусу (рис. 1). Ось OZ направлена продольно от «верхней» части корпуса (направленной от забоя) к «нижней» (направленной в сторону забоя). Ось OY направлена поперечно, от оси прибора в сторону метки на корпусе. Ось OX направлена таким образом, что OX, OY, OZ формируют правый ортонормированный базис.

Направления физических осей чувствительности акселерометров и магнитометров грубо сориентированы в соответствии с направлениями осей прибора. Обозначения $\overline{ig} = \{ig_x, ig_y, ig_z\}$, $\overline{ib} = \{ib_x, ib_y, ib_z\}$ соответствуют сигналу датчиков, полученному контроллером с АЦП, измеряются в условных единицах ускорения и магнитной индукции.

Сигнал $\overline{ig}, \overline{ib}$ подвержен воздействию температурного дрейфа, для которого в программе микроконтроллера производится компенсация, в результате которой получаются сигналы $\overline{tg}, \overline{tb} = thermoCal(\overline{ig}, \overline{ib}, T, \overline{pThermo})$, где T – температура прибора, $\overline{pThermo}$ – параметры температурной калибровки, уникальные для каждого прибора.

Имеет место разброс масштабов, смещений нуля сигнала, обусловленных электроникой, а также обусловленная механически неортогональность осей чувствительности. В микроконтроллере также производится компенсация этих параметров, в результате чего получаются сигналы $\overline{g}, \overline{b} = GeomCal(\overline{tg}, \overline{tb}, \overline{pGeom})$, где $\overline{pGeom}$ – параметры геометрической калибровки, уникальные для каждого прибора.

Калибровочные параметры $\overline{pThermo}, \overline{pGeom}$ вычисляются в процессе настройки прибора при его производстве или ремонте, и записываются при помощи ПО Colibri5 в ПЗУ микроконтроллера.

Углы ориентации вычисляются из сигналов $\overline{g}, \overline{b}$ следующим образом:

$$\boldsymbol{Zeni} = arctan2(\sqrt{g_x^2 + g_y^2}, g_z)$$

$$AzNumer = \sqrt{g_x^2 + g_y^2 + g_z^2} * (g_x b_y - g_y b)$$

$$AzDenom = b_z * \left(g_x^2 + g_y^2\right) - g_z * \left(g_x b_x + g_y b_y\right)$$

$$Azim = (\arctan 2(AzNumer, AzDemon) + 360°) \bmod 360°$$

$$TFG == (\arctan 2(-g_x, g_y) + 360°) \bmod 360°$$

$$TFM == (\arctan 2(-b_x, b_y) + 360°) \bmod 360°$$

$$TF = \begin{cases} TFG, & \text{если } Zeni \geq 4° \\ TFM, & \text{если } Zeni < 4° \end{cases}$$

где *Zeni* – зенитный угол, *Azim* – азимут, *TF* – угол установки отклонителя.

Требуемые погрешности измерения для инклинометра «Луч» аналогичны паспортным погрешностям эксплуатируемого аналога – инклинометра APS760 [2] и приведены в табл. 1.

Таблица 1

Характеристики инклинометра APS760

Table 1

APS760 Inclinometer Specifications

Параметр	**Значение**
Погрешность измерения зенитного угла	±0,1°
Погрешность измерения азимута (при зените 90°)	±0,3°
Погрешность измерения азимута (при зените 10°)	±1°
Погрешность измерения азимута (при зените 5°)	±2°
Погрешность измерения угла установки отклонителя	±0,5°
Диапазон рабочих температур	0…150 °C

Процесс калибровки устраняет значительную часть систематической погрешности, но из-за несовершенства данных и моделей, используемых для калибровки, часть погрешности остается не исключенной, то есть сигналы $\overline{g}, \overline{b}$ содержат в себе отклонение от идеальных проекций. Для соответствия инклинометра заявленным требованиям были определены допустимые

значения такого отклонения. Расчет был произведен программно, для местности с магнитной индукцией 59,08 мкТл и магнитным наклонением 74,79°. При варьировании сигналов $\overline{g}, \overline{b}$ в заданных пределах была определена зависимость вариации углов ориентации от зенита (при всех возможных азимутах и углах установки отклонителя). Результаты расчета представлены на рис. 2. По этим результатам допустимо, например, отклонение в пределах ±0,001 g по всем трем акселерометрам и ±0,20 мкТл по всем трем магнитометрам.

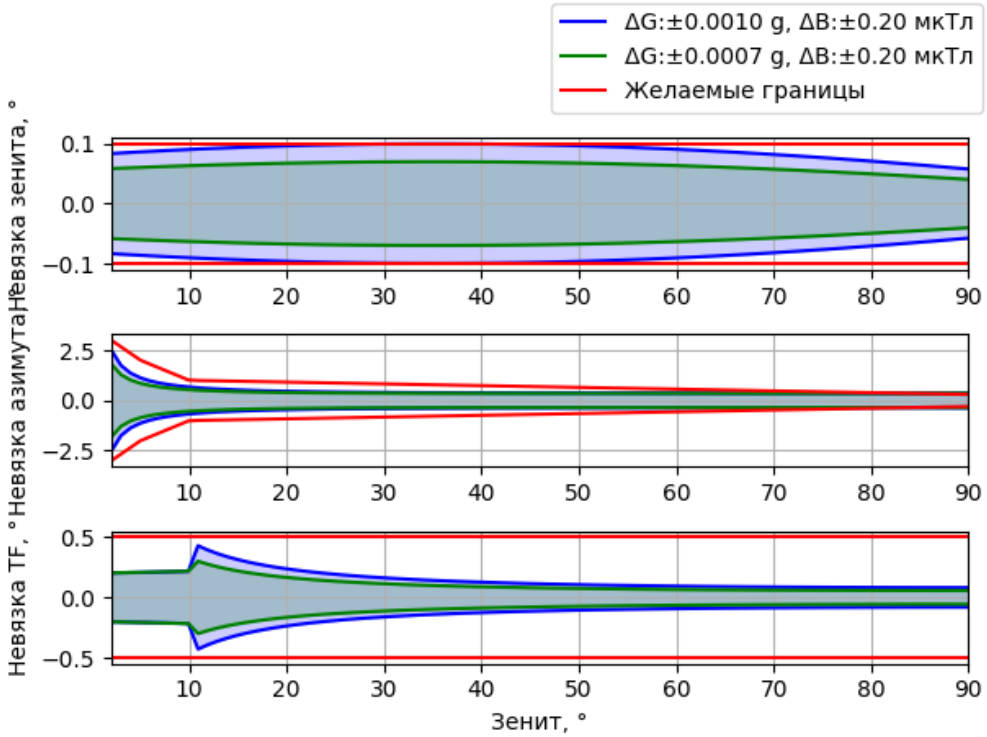

Рис. 2. Вариация углов ориентации при разных вариациях сигналов $\overline{g}, \overline{b}$
Fig. 2. Variation of orientation angles for different variations of signals $\overline{g}, \overline{b}$

К скважинным инклинометрам, применяемым в процессе бурения, предъявляется требование возможности работы в широком диапазоне температур от 0 °С до 120 °С. Изменение рабочей температуры инклинометра ведет к возникновению дрейфа в измерительной части, заметно превышающего допустимые отклонения. Например, при нагреве экспериментального прибора от 25 °С до 140 °С в неизменном положении показания акселерометров изменялись на величины до 0,0155 g, что превышает допустимое отклонение в ±0,001 g, поэтому требует компенсации.

Было принято решение использовать программный метод компенсации дрейфов как наиболее гибкий.

В используемую температурную модель в соответствии с паспортом акселерометров и магнитометров были включены масштабный коэффициент и смещение нуля, зависящие от температуры:

$$m(T) = k(T) * m(T_k) + b(T),$$

где T – температура окружающей среды, °С, $m(T)$ – сигнал датчика при данной температуре, g или мкТл, T_k – комнатная температура, °С, $k(T)$ – масштаб, $b(T)$ – смещение нуля, g или мкТл.

Для определения масштаба и смещения нуля проводился следующий эксперимент. Прибор нагревался закрепленным в наклонной печи (рис. 3), установленной под углом 45° к горизонту, и выполненной из немагнитных материалов, до температуры 150 °C. Затем показания датчиков фиксировались в процессе плавного остывания в течение 12 часов до комнатной температуры. Сигналы регистрировались в течение нескольких циклов нагрева-остывания, на каждой итерации изменялось положение прибора для охвата максимально широкого диапазона значений датчиков (акселерометров и магнитометров).

Рис. 3. Наклонная печь для проведения термоиспытаний, изготовлена из немагнитных материалов
Fig. 3. Inclined furnace for thermal testing, made of non-magnetic materials

Затем вычислялись $k(T)$ и $b(T)$ для каждой из температур в диапазоне 30...150 °C с шагом в 2 °C. Для определенности системы брались пары записей, соответствующих разным положениям прибора.

По результатам экспериментальных циклов нагрева-остывания (рис. 4, 5) функции $k(T)$, $b(T)$ для акселерометров и магнитометров были аппроксимированы многочленами третьей степени. Коэффициенты многочленов записываются в ПЗУ микроконтроллера, и формируют структуру *pThermo*. В процессе работы микропрограммы определяются актуальные значения $k(T)$, $b(T)$ для каждого из 6 датчиков в зависимости от сигнала с термодатчика, размещенного рядом с акселерометрами или магнитометрами, и смоделированный дрейф вычитается из сигнала, поступающего с АЦП.

Также была сформирована методика проведения термокалибровки инклинометра:

1. Запись сигналов датчиков в процессе остывания в двух положениях, широко охватывающих диапазон сигналов датчиков при физических ограничениях печи:

а) зенит = 45°, азимут = 45°, поворот вокруг оси = 45°;

б) зенит = 135°, азимут = 135°, поворот вокруг оси = 225°.

2. Подбор калибровочных параметров на основании данных двух записей; запись полученных настроек в память инклинометра в структуру *pThermo*.

3. Дополнительный цикл нагрева-остывания в третьем положении, для экспериментального подтверждения успешного подбора.

Результаты

По результатам экспериментов температурный дрейф масштабного коэффициента для акселерометров удалось снизить с 1 ± 0,1 до ±0,1 %, дрейф смещения нуля с 0,0012 ± 0,0002 до ±0,0002 *g* на диапазоне температур 20…150 °C (рис. 4).

Дрейф масштабного коэффициента магнитометров удалось снизить с 1,5 ± 0,3 до ±0,3 %, дрейф смещения нуля с 0,1 ± 0,09 до ±0,09 мкТл (рис. 5).

Применение компенсации для каждого из датчиков позволяет снизить дрейф зенита до ±0,12°; азимута до ±0,4° (при зените 90°); угла установки отклонителя до ±0,5°; расчетного модуля ускорения до ±0,0013 *g*; расчетного модуля магнитной индукции с до ±0,32 мкТл.

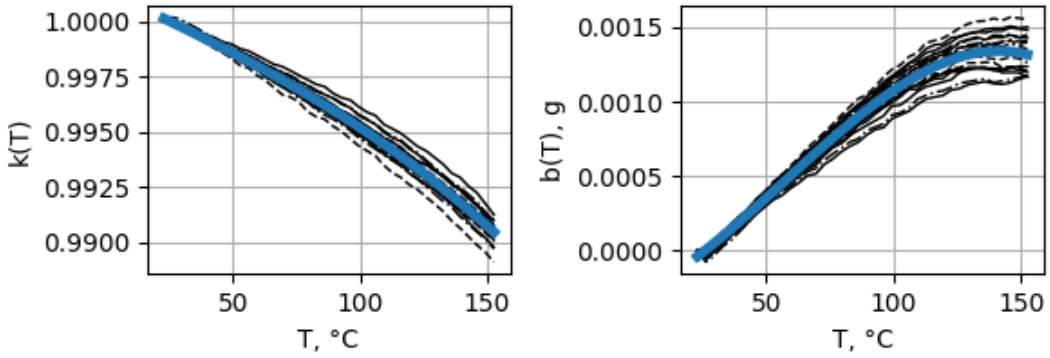

Рис. 4. Характеристики *k*(*T*), *b*(*T*) для одного из трех акселерометров
Fig. 4. Characteristics for one of the three accelerometers

Черными графиками показаны варианты, рассчитанные на основе разных пар записей, со значениями проекций ускорения (*g*) –0,66; –0,5; –0,27; 0,02; 0,49; 0.26; 0,7. Синий график – результат аппроксимации.

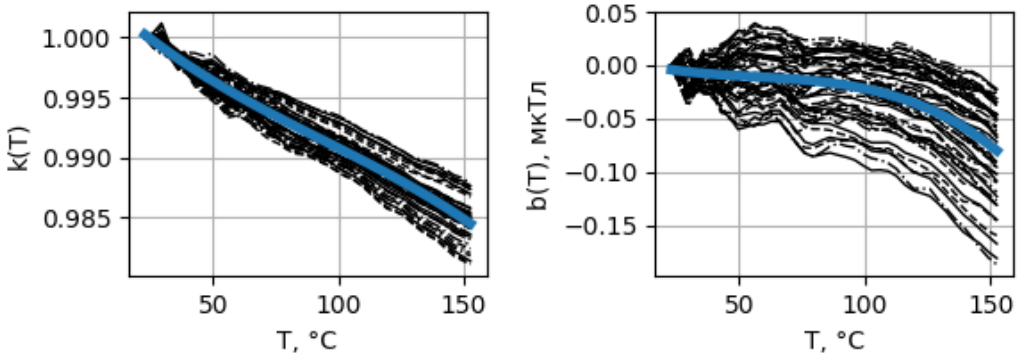

Рис. 5. Характеристики *k*(*T*), *b*(*T*) для одного из трех магнитометров
Fig. 5. Characteristics for one of the three magnetometers

Значения проекции магнитной индукции, для которых проводился расчет (мкТл): –40, –23, –7, 21, 35, 40. Синий график – результат аппроксимации.

В табл. 2 приведен расчет полученной суммарной погрешности в двух температурных диапазонах: 20...120 °С и 20…150 °С.

Таблица 2

Полученные погрешности

Table 2

Obtained errors

Параметры	**Температурная**	**Геометрическая**	**Суммарная**
Зенит, 20..120 °С	±0,085°	±0,01°	±0,095°
Азимут при зените 90°, 20...120 °С	±0,3°	±0,1°	±0,4°
Отклонитель при зените 90°, 20...120 °С	±0,4°	±0,1°	±0,5°
Зенит, 20...150 °С	±0,12°	±0,01°	±0,13°
Азимут при зените 90°, 20...150 °С	±0,4°	±0,1°	±0,5°
Отклонитель при зените 90°, 20…150 °С	±0,5°	±0,1°	±0,6°

Обсуждение

Приведенные в табл. 2 суммарные погрешности в диапазоне температур 20…120 °С превышают заявленные в спецификации APS-760 погрешности по азимутальному углу. Этот факт, возможно, обусловлен разными условиями калибровки. Горизонтальная составляющая магнитного поля в Новосибирске (место калибровки инклинометра «Луч») составляет 15,7 мкТл от общего поля 60,3 мкТл, в Сан-Франциско (место калибровки APS-760) составляет 22,9 мкТл от общего поля 47,5 мкТл. Погрешность определения азимутального угла напрямую зависит от модуля горизонтальной составляющей магнитного поля Земли. Остальные суммарные погрешности не превышают аналогичные у APS760.

Вне зависимости от температурного диапазона полученные погрешности соответствуют руководящему документу: «Допускаемая основная погрешность измерения азимута для зенитных углов более 3° не более ±2°, допускаемая основная погрешность измерения зенитного угла – не более ±0,5°» [6]. Корпоративные стандарты заказчиков предъявляют более строгие требования к характеристикам прибора, чем руководящий документ: допускаемая основная погрешность измерения азимута для зенитных углов более 3°, не более ±1,5°, допускаемая основная погрешность измерения зенитного угла – не более ±0,25°. Полученные погрешности соответствуют и этим стандартам.

Заключение

Разработано программное обеспечение, методика калибровки инклинометра производства ООО НПП ГА «Луч», проведены лабораторные испытания с положительным результатом. Благодаря переходу на ПО и платы собственного производства ожидается повышение ремонтопригодности и снижение трудозатрат на модернизацию оборудования.

Список литературы

1. **Ковшов Г. Н., Коловертнов Г. Ю.** Приборы контроля пространственной ориентации скважин при бурении. Уфа: Изд-во УГНТУ, 2001. 228 с.

2. Model 760 Directional Sensor // Applied Physics Systems. URL: https://appliedphysics.com/wp-content/uploads/2022/06/ APS_DataSheet_Model760_vA.pdf (дата обращения: 17.01.2024).
3. Семейство универсальных гироскопических инклинометров УГИ-42 // АО «СКБ ПН». URL: http://skbpn.ru/ugi-42 (дата обращения: 17.04.2024).
4. Каротажный кабельный магнитометрический инклинометр «Кварц-36.04К» // АО «СКБ ПН». URL: http://skbpn.ru/quartz-36.04 (дата обращения: 17.04.2024).
5. Свидетельство о государственной регистрации программы для ЭВМ №2018664639. Colibri5 / А. А. Власов, Д. В. Тейтельбаум, А. М. Найденов – Заявка №2018618675. Дата поступления 14 августа 2018 г. Зарегистрировано в Реестре программ для ЭВМ 20 ноября 2018 г.
6. РД 153-39.0-072-01 «Техническая инструкция по проведению геофизических исследований и работ приборами на кабеле в нефтяных и газовых скважинах» (введен в действие приказом Министерства энергетики РФ от 7 мая 2001 г. N 134)

References

1. **Kovshov G. N., Kolovertnov G. Yu.** Instruments for monitoring the spatial orientation of wells during drilling. Ufa, Publishing house of Ufa State Petroleum Technical University, 2001, 228 p.
2. Model 760 Directional Sensor. *Applied Physics Systems*. URL: https://appliedphysics.com/wp-content/uploads/2022/06/ APS_DataSheet_Model760_vA.pdf (date of access: 17.01.2024).
3. Family of universal gyroscopic inclinometers UGI-42. *JSC "SKB PN"*. URL: http://skbpn.ru/ugi-42 (date of access: 17.04.2024).
4. Logging cable magnetometric inclinometer "Quartz-36.04K". *JSC "SKB PN"*. URL: http://skbpn.ru/quartz-36.04 (date of access: 17.04.2024).
5. Certificate of state registration of computer program No. 2018664639. Colibri5 / A. A. Vlasov, D. V. Teitelbaum, A. M. Naidenov – Application No. 2018618675. Date of receipt: August 14, 2018. Registered in the Register of Computer Programs: November 20, 2018
6. RD 153-39.0-072-01 "Technical instructions for conducting geophysical surveys and work with cable instruments in oil and gas wells" (put into effect by order of the Ministry of Energy of the Russian Federation dated May 7, 2001. N 134).

Сведения об авторах

Литвинов Василий Сергеевич, инженер-программист

Власов Александр Александрович, кандидат технических наук, ведущий геофизик
Research ID: J-3644-2018

Тейтельбаум Дмитрий Владимирович, начальник отдела программного обеспечения

Information about the Authors

Vasily S. Litvinov, Software Engineer

Alexander A. Vlasov, Ph.D., Leading Geophysicist, Engineer
Research ID: J-3644-2018

Dmitry V. Teytelbaum, Head of Software Department

Статья поступила в редакцию 15.10.2024;
одобрена после рецензирования 28.10.2024; принята к публикации 28.10.2024

The article was submitted 15.10.2024;
approved after reviewing 28.10.2024; accepted for publication 28.10.2024

Научная статья

УДК 004.912 + 004.8
DOI 10.25205/1818-7900-2024-22-3-49-61

Сравнение методов машинного обучения для решения задачи анализа тональности

Милана Владимировна Швенк[1], Елена Павловна Бручес[1, 2]
Анна Яковлевна Леман[1]

[1]Новосибирский государственный университет
Новосибирск, Россия

[2]Институт систем информатики им. А. П. Ершова СО РАН
Новосибирск, Россия

m.shvenk@g.nsu.ru
bruches@bk.ru
a.leman@g.nsu.ru

Аннотация

Сеть «Интернет» позволяет делиться своим мнением со всем миром, и эти данные используются для анализа отношения общества к тому или иному субъекту. С течением времени эмоционально окрашенных текстов становится все больше. Современные технологии позволяют обрабатывать огромные массивы данных в кратчайшие сроки, что по-настоящему важно при реализации анализа тональности, который является одним из актуальных направлений автоматической обработки естественного языка. Нами был собран и размечен корпус текстов отзывов на медицинские услуги. Также были апробированы три способа решения задачи анализа тональности, относящиеся к методам традиционного или глубокого машинного обучения. Проведен сравнительный анализ полученных результатов. Размеченный нами корпус выложен в открытый доступ и может быть использован для других исследований.

Ключевые слова

обработка естественного языка, анализ тональности, машинное обучение, логистическая регрессия, сверточная нейронная сеть, LSTM

Для цитирования

Швенк М. В., Бручес Е. П., Леман А. Я. Сравнение методов машинного обучения для решения задачи анализа тональности // Вестник НГУ. Серия: Информационные технологии. 2024. Т. 22, № 3. С. 49–61. DOI 10.25205/1818-7900-2024-22-3-49-61

Comparison of Machine Learning Methods for Sentiment Analysis

Milana V. Shvenk[1], Elena P. Bruches[1, 2]
Anna Y. Leman[1]

[1]Novosibirsk State University,
Novosibirsk, Russian Federation

[2]A. P. Ershov Institute of Informatics Systems SB RAS,
Novosibirsk, Russian Federation

m.shvenk@g.nsu.ru
bruches@bk.ru
a.leman@g.nsu.ru

Abstract

Every day the amount of text data containing the subjective evaluation of the author is increasing thanks to the Internet. This information is used, for example, by numerous companies to assess the loyalty of their target audience. Due to the incredibly fast growth of the volume of such texts, their manual processing becomes impractical. It is in such situations that automated sentiment analysis is used, which is an actively developing area of natural language processing. We collected a corpus of medical service reviews, on the basis of which three classifiers were trained. We also performed a comparative analysis of the obtained results of the models, which belong to traditional or deep machine learning. Our corpus of texts is public and can be useful for other researchers.

Keywords

natural language processing (NLP), sentiment analysis, machine learning, logistic regression, convolutional neural network (CNN), LSTM

For citation

Shvenk M. V., Bruches E. P., Leman A. Y. Comparison of machine learning methods for sentiment analysis. *Vestnik NSU. Series: Information Technologies*, 2024, vol. 22, no. 3, pp. 49–61 (in Russ.) DOI 10.25205/1818-7900-2024-22-3-49-61

Введение

Большинство людей всегда интересуются точкой зрения своих знакомых для принятия решений, укрепления собственной позиции, сравнения взглядов и многого другого. Развитие сети «Интернет», а также ее возросшая доступность для обычного человека позволяют узнавать мнения и опыт людей, которые не являются ни нашими знакомыми, ни профессиональными критиками, ни даже жителями одной с нами страны. Соответственно, все больше людей делятся собственным мнением, увеличивая объем подобной информации, находящейся в открытом доступе.

Эти данные используются компаниями для определения лояльности потребителей к своему бренду/продукции/услугам; должностными лицами органов государственной власти [1]; инвесторами для выбора стратегии инвестирования [2] и т. д.

Анализ тональности текста – одно из актуальных направлений автоматической обработки естественного языка.

Человек способен прочитать несколько текстов и определить их тональность, в то время как программа за это же время обработает тысячи текстов, хоть и с относительно меньшей точностью.

В данной статье рассмотрены три способа решения задачи анализа тональности, принадлежащие традиционному и глубокому машинному обучению. Также проведено сравнение полученных результатов. Обучение и тестирование наших классификаторов проводилось на собранном и размеченном нами корпусе, который доступен по ссылке: https://github.com/m-sh-v/Sentiment_analysis/tree/main/corpus.

Обзор подходов к реализации анализа тональности

Основными методами решения задачи анализа тональности являются следующие три подхода.

Лингвистический. Используя словари эмоционально окрашенной лексики [3], подсчитывается количество пересечений их данных со словами обрабатываемого текста. После чего тексту присваивается тональность в зависимости от количества позитивной и негативной лексики. Однако данный метод имеет весомые недостатки [4]: качественное составление словаря требует огромных временных затрат; также автор-составитель должен разбираться в предметной области; один и тот же словарь нельзя использовать на разных предметных областях без потерь качества классификации.

Использование машинного обучения. Предобученная на заранее подготовленных данных модель используется для классификации новых текстов [5]. При выборе данного метода качество подготовки обучающих данных напрямую влияет на эффективность модели. Машинное обучение подразумевает под собой множество алгоритмов и способов решения задач и хорошо себя зарекомендовало при осуществлении анализа тональности [6; 7].

Гибридный. Совмещение двух вышеописанных методов, являясь наиболее трудоемким и ресурсозатратным, позволяет увеличить качество работы классификатора [8].

Нами были апробированы подходы на основе машинного обучения (сверточная нейронная сеть[1], LSTM) и гибридный метод (логистическая регрессия).

Составление и разметка собственного корпуса

Тематикой текстов нашего корпуса стали отзывы на медицинские услуги. При выборе направленности отзывов мы исходили из факта частого использования ярко эмоционально окрашенной лексики, а также мы старались избежать проблемы мультилингвальности [8], которая проявляется особенно сильно при обработке данных, взятых из социальных сетей. Например, анализ данных могут осложнять хештеги, написанные на языке, отличном от языка основного текста, а также хештег на одном языке может использоваться в постах носителей разных языков.

Сбор материала проводился с использованием открытого веб-ресурса «infodoctor»[2], который содержит обширную базу отзывов, каждому из которых автором комментария присвоена субъективная оценка (от 1 до 5). В нашей работе мы отбирали комментарии с оценками «1» и «5», которые содержат большее количество эмоционально окрашенной лексики. Выбранная нами предметная область позволяет относить даже условно нейтральные отзывы к классу позитивных, например, комментарий «Отправили на анализы. Сделали УЗИ», которому была присвоена оценка «5». Таким образом, для решения нашей задачи, т. е. определения тональности отзывов на медицинские услуги, мы можем ограничиться бинарной классификацией (негативный/позитивный).

Мы стремились сделать разметку нашего корпуса максимально полной, предвосхищая возможность его использования для решения задач из других сфер. При этом стоит отметить, что при обучении наших моделей мы использовали не все атрибуты разметки (см. табл. 1), исходя из специфики задачи анализа тональности.

Описание атрибутов корпуса: «text» – текст комментария; «doc» – отзыв на доктора (1 – да, 0 – нет); «service» – отзыв на услугу (1 – да, 0 – нет); «stars» – субъективная оценка автора комментария (от 1 до 5); «pos» – «позитивный» (1 – да, 0 – нет); «neu» – «нейтральный» (1 – да, 0 – нет); «neg» – «негативный» (1 – да, 0 – нет); «score» – объединение трех предыдущих

[1] Convolutional neural network (CNN).

[2] Сервис подбора врача и онлайн-записи на прием «ИнфоДоктор» основан в сентябре 2011 г., впервые анонсирован в 2012 г. (https://infodoctor.ru/).

атрибутов (1/0/–1); «area» – город расположения медицинского учреждения («spb» – Санкт-Петербург, «msk» – Москва, «ekb» – Екатеринбург, «nsk» – Новосибирск)

Таблица 1

Пример разметки корпуса на основе текста одного комментария

Table 1

An example of corpus markup based on the text of a comment

texts	doc	service	stars	pos	neu	neg	score	area
Ужасный косметолог, делала тредлифтинг и что в итоге, испортила мне лицо, кормит завтраками что скоро пройдет, как работать, как выходить на улицу, с мужем конфликт.	1	0	1	0	0	1	-1	spb

Всего было собрано 11779 текстов отзывов, 5899 из которых являются негативными, 5880 – позитивными. Все данные корпуса для обучения классификаторов были представлены в формате xml.

Впоследствии общее количество текстов было разделено на обучающую и тестовую выборки (см. табл. 2) методом удерживания[3], который подходит для большого объема данных [9].

Таблица 2

Распределение текстов корпуса

Table 2

Dataset samples distribution

	Обучающая выборка	Тестовая выборка
Негативные отзывы	5230	669
Позитивные отзывы	5186	694
Общее количество текстов	10416	1363

Качество разметки проверялось одной из наших моделей, т. е. мы использовали ее для предсказания классов тех же данных, на которых она была обучена: если присвоенная нами тональность текста не совпадала с тональностью, которую предсказала наш классификатор, то атрибуты данного текста отзыва были перепроверены вручную. Таким образом, было выявлено около 30 отзывов, которые подверглись исправлению. Ошибки при разметке были совершены из-за неправильно выставленной оценки авторов своим отзывам, т. е. человеческого фактора. Соответственно, после данной проверки качество разметки нашего корпуса возросло.

[3] Holdout. Подход состоит в разделении исходного набора данных случайным образом на два непересекающихся множества.

Традиционное машинное обучение и глубокое машинное обучение

Схематичное представление традиционного машинного обучения изображено на рис. 1.

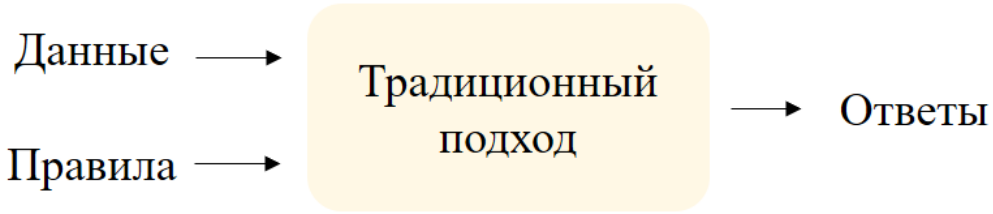

Рис. 1. Схема традиционного машинного обучения
Fig. 1. Scheme of traditional machine learning

Не все возникающие задачи возможно решить традиционными методами, которые основываются на правилах. Тогда в дело вступает глубокое машинное обучение [10].

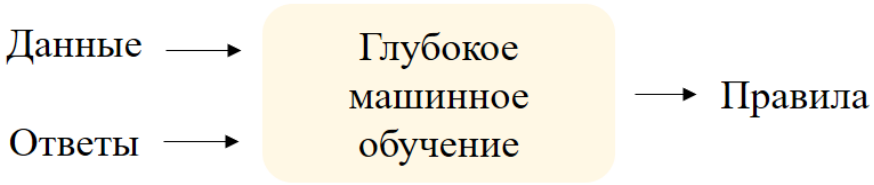

Рис. 2. Схема глубокого машинного обучения
Fig. 2. Scheme of deep machine learning

Пусть схема глубокого машинного обучения (см. рис. 2) и похожа на схему традиционного, ключевым ее отличием является выделение признаков (правил) без участия человека.

Этап предобработки текстов

Компьютер считывает информацию не так, как это делает человек, поэтому необходимо произвести предварительную подготовку данных, т. е. препроцессинг [11], который в нашем случае содержит следующие этапы обработки данных: очистка от числовых и специальных символов, которые не несут значения для классификации; удаление стоп-слов[4] [12]; понижение регистра [13]; токенизация[5] [14] (для нас токеном является слово); лемматизация[6], которая позволяет снизить нагрузку на классификатор [15]; векторизация[7] [14].

Также мы выбрали одну из ведущих библиотек нейронных сетей – Keras[8]. Данная библиотека позволяет проводить эксперименты с моделями нейронных сетей, обладая возможностью

[4] Стоп-слова – слова, не несущие важного семантического смысла для модели, например, служебные части речи, артикли, междометия, союзы. В русском языке к таким словам относятся: «и», «вы», «ах», «за» и т .п.

[5] Токенизация – разбиение строки на слова, фразы или символы.

[6] Лемматизация – приведение слов к их начальной форме.

[7] Векторизация – преобразование текстовых данных в числовые векторы.

[8] https://keras.io.

сократить время и усилия при прототипировании, так как наличие обширной базы уже созданных слоев, оптимизаторов и функций потерь дает возможность не тратить время на кодирование всех настроек вручную [16].

Логистическая регрессия

Наилучшие результаты среди методов традиционного машинного обучения показывают линейные модели, к которым относятся логистическая регрессия и метод опорных векторов [4]. В качестве представителя традиционного машинного обучения нами была выбрана логистическая регрессия, так как в разных исследованиях она показывает более высокие результаты [14; 17; 18].

Для выделения признаков принадлежности текста к определенному классу [19] мы использовали два метода: 1) подсчитывание количества эмоционально окрашенной лексики с помощью готового словаря RuSentiLex (РуСентиЛекс), записи в котором содержат ссылки на понятия РуТез[9]; 2) оценивание значимости слова в документе, т. е. метод TF-IDF[10], из которого следует, что токен, наиболее распространенный в одном тексте, но менее – в остальных, получает больший вес [20].

При использовании первого метода мы придерживались правила: если количество негативной лексики преобладало – отзыв негативный, во всех других случаях – отзыв позитивный.

После применения второго метода было выделено 20 % самых значимых токенов по результатам критерия хи-квадрата Пирсона.

На основе полученных признаков мы обучили классификатор и провели оценку качества его работы (табл. 3).

Таблица 3

Результаты оценки качества работы классификатора, основанного на логистической регрессии

Table 3

Logistic regression based classifier evaluation results

	precision	**recall**	**f1-score11**
Negative	0,9273	0,9925	0,9588
Positive	0,9923	0,9250	0,9575
		Weighted Average[12]	0,9582

Результаты показывают, что:

1) 93 % текстов, которые наша модель определила как негативные, действительно являются таковыми. Модель распознала 99 % всех негативных отзывов;

2) модель распознала 93 % всех позитивных текстов. Из них 99 % действительно оказались позитивными;

3) общая оценка эффективности классификатора составляет 96 %.

[9] Лингвистическая онтология РуТез (англ. RuThes) – тезаурус русского языка, представляющий собой иерархическую сеть понятий.

[10] Term frequency-inverse document frequency.

[11] F1-score – гармоническое среднее.

[12] Weighted average – тип среднего значения, придающий различную важность значениям в наборе данных.

Сверточная нейронная сеть

Наиболее зарекомендовавшими себя в исследуемой нами области методами, относящимися к глубокому машинному обучению, являются сверточные и рекуррентные[13] нейронные сети [4]. В нашей статье мы рассматриваем оба типа.

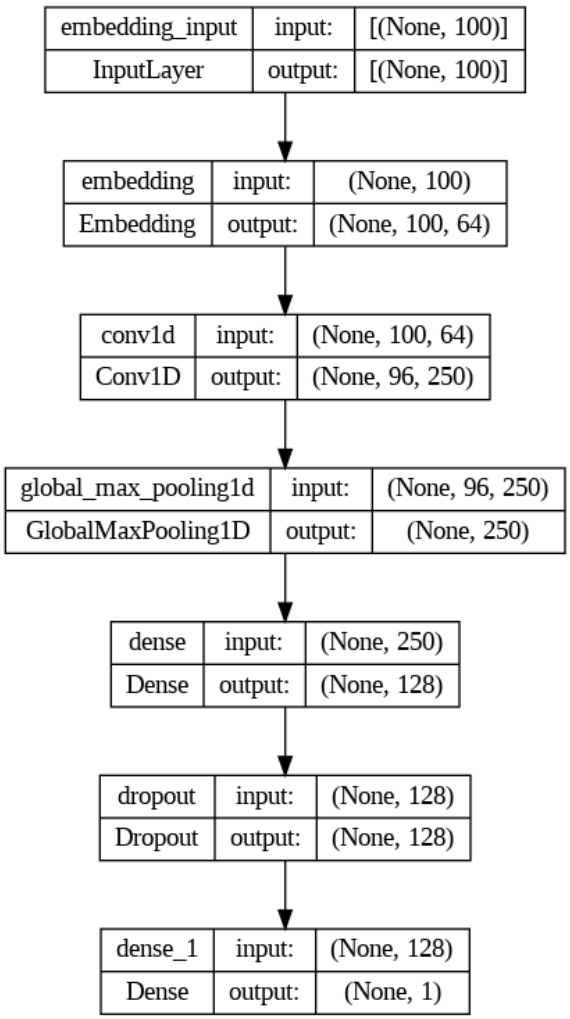

Рис. 3. Архитектура сверточной нейронной сети
Fig. 3. Convolutional neural network architecture

Архитектура нашей сверточной нейронной сети представлена на рис. 3. Наша модель состоит из шести слоев: embedding-слой, сверточный слой, субдискретизирующий слой, полносвязный слой, dropout-слой и еще один полносвязный слой.

При обучении нашего классификатора лучшие результаты были показаны на третьей эпохе обучения (табл. 4), поэтому мы сохранили модель с ее лучшими показателями и использовали именно ее в дальнейшем.

[13] Recurrent neural network (RNN).

Таблица 4

Показатели качества на каждой эпохе обучения классификатора на основе сверточной нейронной сети

Table 4

Quality scores at each epoch of CNN-based classifier training

Эпоха	**train_accuracy**[14]	**val_accuracy**[15]
1	0,8032	0,9155
2	0,9713	0,9702
3	0,9941	0,9731
4	0,9994	0,9568

После проверки классификатора на тестовой выборке мы получили результаты, приведенные в табл. 5.

Таблица 5

Результаты оценки качества работы классификатора, основанного на сверточной нейронной сети

Table 5

CNN-based classifier evaluation results

	precision	**recall**	**f1-score**
Negative	0,9719	0,9821	0,9770
Positive	0,9825	0,9726	0,9775
		Weighted Average	**0,9773**

Данные результаты показывают, что:

1) 97 % текстов, которые наша модель назвала негативными, действительно являются негативными; модель распознала 98 % от всего количества негативных отзывов;

2) модель распознала 97 % от общего количества позитивных текстов; 98 % комментариев, которые модель определила позитивными, действительно такими являются;

3) общая оценка эффективности классификатора равна 98 %.

LSTM

Также в нашей статье мы рассматриваем еще один тип нейронной сети – рекуррентный, который по многочисленным исследованиям является одним из лучших решений в вопросе обработки естественного языка [21].

Подобные нейронные сети способны использовать свою внутреннюю память для обработки серии событий и последовательностей произвольной длины [22]. Наличие обратных связей особенно важно для анализа тональности, поскольку текст необходимо рассматривать именно как последовательность [21].

[14] Показатели accuracy на обучающем наборе данных в процессе обучения.

[15] Показатели accuracy на валидационном наборе данных в процессе обучения.

Как бы хорошо все ни звучало, рекуррентные сети имеют весомый недостаток, названный проблемой исчезающего градиента[16] [23].

Именно для борьбы с данным явлением была создана особая архитектура рекуррентной нейронной сети – LSTM[17] [24].

Наша модель имеет четыре слоя (рис. 4): embedding-слой, два lstm-слоя и полносвязный слой.

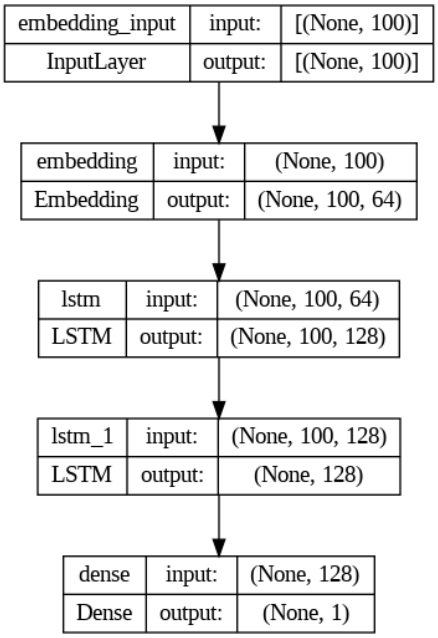

Рис. 4. Архитектура модели LSTM
Fig. 4. LSTM model architecture

Сохранив модель с ее лучшими показателями (табл. 6) в процессе обучения, мы провели оценку ее работы на тестовом наборе данных.

Таблица 6

Показатели качества на каждой эпохе обучения классификатора на основе LSTM

Table 6

Quality scores at each epoch of LSTM-based classifier training

Эпоха	train_accuracy	val_accuracy
1	0,8462	0,9280
2	0,9743	0,9530
3	**0,9931**	**0,9655**
4	0,9984	0,9530
5	0,9998	0,9616

После проверки модели на тестовой выборке мы получили результаты, приведенные в табл. 7.

[16] Vanishing gradients problem.

[17] Long Short Term Memory – «Долгая краткосрочная память».

Таблица 7

Результат оценки качества модели LSTM

Table 7

LSTM-based classifier evaluation results

	precision	**recall**	**f1-score**
Negative	0,9836	0,9821	0,9828
Positive	0,9827	0,9841	0,9834
		Weighted Average	**0,9831**

Данные результаты показывают:

1) 98 % текстов, которые наша модель назвала негативными, действительно являются негативными; модель распознала 98 % от всего количества негативных отзывов;

2) модель распознала 98 % от общего количества позитивных текстов; 98 % комментариев, которые модель определила позитивными, действительно такими являются;

3) общая оценка эффективности классификатора равна 98 %.

Анализ результатов

По результатам апробации трех вариантов решения задачи анализа тональности, основанных на применении традиционных и глубоких методов машинного обучения, были получены следующие результаты.

Представитель традиционного машинного обучения, логистическая регрессия, на которой основана наша первая модель, по оценке f1-score показала эффективность классификатора, равную 96 %.

Методы глубокого машинного обучения показали следующие результаты: классификатор, основанный на сверточной нейронной сети – 98 %; классификатор, построенный с использованием lstm-слоев – 98 %.

Стоит учесть, что оба классификатора, основанных на методах глубокого машинного обучения, показывают результат, равный 98 %, но при этом сверточная нейронная сеть подходит к данному показателю снизу (0,9773), в то время как модель LSTM – сверху (0,9831).

Заключение

В ходе работы мы собрали и разметили корпус, состоящий из текстов отзывов на медицинские услуги на русском языке. Мы описали используемые нами подходы к решению задачи анализа тональности, отметив их преимущества и отличия. Апробировали три способа реализации данного анализа и оценили качество работы классификаторов, каждый из которых можно использовать на практике для быстрой оценки большого объема данных, а конкретно для проведения анализа тональности отзывов на медицинские услуги, что позволит медицинским учреждениям отслеживать отношение пациентов к качеству предоставляемых услуг.

Таким образом, наша гипотеза подтвердилась: глубокое машинное обучение превосходит традиционный подход при решении задачи анализа тональности, а LSTM оказалась более эффективной моделью, чем другой представитель глубокого машинного обучения – сверточная нейронная сеть.

Список литературы

1. **Андросов А. Ю., Бородащенко А. Ю., Леонидова К. С.** Алгоритм определения тональности публикаций СМИ к должностным лицам государственных органов // Известия ТулГУ. Технические науки. 2020. № 2. С. 47–53.
2. **Ксенофонтов Г. С.** Тональный анализ новостей экономики // Скиф. Вопросы студенческой науки. 2022. № 5 (69). С. 584–589.
3. **Пазельская А. Г., Соловьев А. Н.** Метод определения эмоций в текстах на русском языке // Компьютерная лингвистика и интеллектуальные технологии: Тр. Междунар. конференции «Диалог'2014». М.: Наука, 2014. С. 574–586.
4. **Самигулин Т. Р., Джурабаев А. Э. У.** Анализ тональности текста методами машинного обучения // Научный результат. Информационные технологии. 2021. Т. 6, № 1. С. 55–62. DOI: 10.18413/2518-1092-2021-6-1-0-7
5. **Пескишева Т. А.** Методы анализа тональности текстов на естественном языке // Общество. Наука. Инновации (НПК-2017). 2017. С. 1730–1742.
6. **Астапов Р. Л., Дубатов Р. С.** Классификация текстов с помощью сверточных нейронных сетей // Вестник науки. 2020. № 8 (29). С. 53–56.
7. **Воробьев Н. В., Пучков Е. В.** Классификация текстов с помощью сверточных нейронных сетей // Молодой исследователь Дона. 2017. № 6 (9). С. 2–7.
8. **Бодрунова С. С.** Кросс-культурный тональный анализ пользовательских текстов в Твиттере // Вестник Моск. ун-та. Серия 10. Журналистика. 2018. № 6. С. 191–212. DOI: 10.30547/vestnik.journ.6.2018.191212
9. **Шунина Ю. С.** Влияние способа формирования обучающей и тестовой выборок на качество классификации // Вестник УлГТУ. 2015. № 2 (70). С. 43–46.
10. **Скрипачев В. О., Гуйда М. В., Гуйда Н. В., Жуков А. О.** Особенности работы сверточных нейронных сетей // International Journal of Open Information Technologies. 2022. № 12. С. 5361.
11. **Майоров Д. В.** Применение глубокого обучения в предиктивном вводе // Научный журнал. 2023. № 2 (67). С. 19–25.
12. **Muhamediyeva D. K., Abdurakhmanova N. N., Mirzayeva N. S.** Using conventional neural networks for the problem of text classification // Central Asian Academic Journal of Scientific Research. 2021. No. 1. p. 213–220.
13. **Abinash Tripathy.** Sentiment Analysis Using Machine Learning Techniques: dissertation // Department of Computer Science and Engineering National Institute of Technology Rourkela. 2017. p. 131.
14. **Бородин А. И., Вейнберг Р. Р., Литвишко О. В.** Методы обработки текста при создании чат-ботов // Хуманитарни Балкански изследвания. 2019. № 3 (5). С. 108–111. DOI: 10.34671/SCH.HBR.2019.0303.0026
15. **Сафаров Л. С.** Использование технологии text mining при автоматической обработке текста // Экономика и социум. 2023. № 1−2 (104). С. 639–642.
16. **Бербасов В. Д.** Сравнительный обзор библиотек нейронных сетей Keras и Pytorch // Экономика и социум. 2023. № 8 (111). С. 423–426.
17. **Васильченко А. М.** Решение задач анализа данных на основе машинного обучения // Universum: технические науки. 2023. № 9−1 (114). С. 50–54. DOI: 10.32743/UniTech.2023.114.9.15959
18. **Шунина Ю. С., Алексеева В. А., Клячкин В. Н.** Критерии качества работы классификаторов // Вестник УлГТУ. 2015. № 2 (70). С. 67–70.
19. **Bing Liu.** Sentiment Analysis and Opinion Mining // Morgan & Claypool Publishers. 2012. P. 168.

20. **Ткаченко А. Л.** Решение задачи классификации документов вуза на основе методов интеллектуального анализа // Вестник кибернетики. 2021. № 1 (41). С. 12–19. DOI: 10.34822/1999-7604-2021-1-12-19
21. **Гальченко Ю. В., Нестеров С. А.** Классификация текстов по тональности методами машинного обучения // SAEC. 2023. № 3. С. 369–378. DOI: 10.18720/SPBPU/2/id23-501
22. **Юркина А. В., Крутиков А. К.** Разработка специализированного программного модуля формирования обучающей выборки для нейрогороскопа // Universum: технические науки. 2023. № 10-1 (115). С. 61–65.
23. **Hochreiter S., Bengio Y., Frasconi P. Schmidhuber J.** Gradient flow in recurrent nets: the difficulty of learning long-term dependencies // IEEE Press. 2001. no. 1. p. 15.
24. **Пустынный Я. Н.** Решение проблемы исчезающего градиента с помощью нейронных сетей долгой краткосрочной памяти // Инновации и инвестиции. 2020. № 2. С. 130–132.

References

1. **Androsov A. Y., Borodaschenko A. Y., Leonidova K. S.** Algorithm for determining the tone of media publications to public officials. *Izvestia TulSU. Engineering Sciences*, 2020, no. 2, pp. 4753. (in Russ.)
2. **Ksenofontov G. S.** Tonal analysis of economic news. *Skif. Student Science Issues,* 2022, no. 5 (69), pp. 584–589. (in Russ.)
3. **Pazelskaya A. G., Solovyov A. N.** The method of determining emotions in texts in Russian. *Computer Linguistics and Intelligent Technologies: Proceedings of the International Conference Dialogue'2014*. Moscow, Science publ., 2014, pp. 574–586. (in Russ.)
4. **Samigulin T. R. Djurabaev A. E. U.** Sentiment analysis of text by machine learning methods. *Research Result. Information Technologies*, 2021, vol. 6, no. 1, pp. 55–62. (in Russ.) DOI: 10.18413/2518-1092-2021-6-1-0-7
5. **Peskisheva T. A.** Methods for sentiment analysis of natural language texts. *Society. Science. Innovations (Scientific and practical conference 2017)*, 2017, pp. 1730–1742. (in Russ.)
6. **Astapov R. L., Dubatov R. S.** Classification of texts using convolutional neural networks. *Vestnik of Science*, 2020, no. 8 (29), pp. 53–56. (in Russ.)
7. **Vorobev N. V., Puchkov E. V.** Classification of texts using convolutional neural networks. *Molodoj issledovatel' Dona*, 2017, no. 6 (9), pp. 2–7. (in Russ.)
8. **Bodrunova S. S.** Cross-cultural sentiment analysis of user texts on twitter. *Vestnik of Moscow University. Series 10. Journalism*, 2018, no. 6, pp. 191–212. (in Russ.) DOI: 10.30547/vestnik.journ.6.2018.191212
9. **Shunina U. S.** Dependence of classification quality on the methods to form a training and test samples. *Vestnik UlSTU*, 2015, no. 2 (70), pp. 43–46. (in Russ.)
10. **Skripachyov V. O., Guyda M. V., Guyda N. V., Zhukov A. O.** Features of convolutional neural networks. *International Journal of Open Information Technologies*, 2022, no. 12, pp. 53–61. (in Russ.)
11. **Mayorov D. V.** Applying deep learning in predictive input. *Scientific Journal*, 2023, no. 2 (67), pp. 19–25. (in Russ.)
12. **Muhamediyeva D. K., Abdurakhmanova N. N., Mirzayeva N. S.** Using conventional neural networks for the problem of text classification. *Central Asian Academic Journal of Scientific Research*, 2021, no. 1, pp. 213–220.
13. **Abinash Tripathy.** Sentiment Analysis Using Machine Learning Techniques: dissertation. *Department of Computer Science and Engineering National Institute of Technology Rourkela*, 2017, p. 131.

14. **Borodin A. I., Veynberg R. R., Litvishko O. V.** Methods of text processing when creating chatbots. *Humanitarian Balkan Studies*, 2019, no. 3 (5), pp. 108–111. (in Russ.) DOI: 10.34671/SCH.HBR.2019.0303.0026
15. **Saifarov L. S.** Using text mining technology in automatic text processing. *Economics and Society*, 2023, no. 1−2 (104), pp. 639–642. (in Russ.)
16. **Berbasov V. D.** Comparative review of keras and pytorch neural network libraries. Economics and Society, 2023, no. 8 (111), pp. 423–426. (in Russ.)
17. **Vasilchenko A. M.** Solving data analysis problems based on machine learning. *Universum: Engineering Sciences*, 2023, no. 9−1 (114), pp. 50–54. (in Russ.) DOI: 10.32743/UniTech.2023.114.9.15959
18. **Shunina U. S., Alekseeva V. A., Klyachkin V. N.** Criteria of quality of qualifiers work. *Vestnik UlSTU*, 2015,no. 2 (70), pp. 67–70. (in Russ.)
19. **Bing Liu.** Sentiment Analysis and Opinion Mining. Morgan & Claypool Publishers, 2012, p. 168.
20. **Tkachenko A. L.** Solving the problem of university documents classification based on intellectual analysis methods. *Vestnik of Cybernetics*, 2021, no. 1 (41), pp. 12–19. DOI: 10.34822/1999-7604-2021-1-12-19 (in Russ.)
21. **Galchenko Y. V., Nesterov S. A.** Sentiment analysis with machine learning methods. SAEC, 2023, no. 3, pp. 369–378. DOI: 10.18720/SPBPU/2/id23-501 (in Russ.)
22. **Yurkina A. V., Krutikov A. K.** Development of a specialized software module for the formation of a training sample for a neurogoroscope. *Universum: Engineering Sciences*, 2023, no. 10−1 (115), pp. 61–65. (in Russ.)
23. **Hochreiter S., Bengio Y., Frasconi P., Schmidhuber J.** Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. *IEEE Press*, 2001, no. 1, p. 15.
24. **Pustynnyj I. N.** Solving the problem of vanishing gradient using long short term memory neural networks. *Innovations and Investments*, 2020, no. 2, pp. 130–132. (in Russ.)

Сведения об авторах

Швенк Милана Владимировна, бакалавр

Бручес Елена Павловна, младший научный сотрудник, старший преподаватель

Леман Анна Яковлевна, старший преподаватель

Information about the Authors

Milana V. Shvenk, Bachelor

Elena P. Bruches, Junior Researcher; Senior Lecturer

Anna Y. Leman, Senior Lecturer

Статья поступила в редакцию 16.07.2024;
одобрена после рецензирования 08.10.2024; принята к публикации 08.10.2024

The article was submitted 1.07.2024;
approved after reviewing 08.10.2024; accepted for publication 08.10.2024

Правила оформления текста рукописи

Авторы представляют статьи на русском или английском языке объемом от 0,5 авторского листа (20 тыс. знаков) до 1 авторского листа (40 тыс. знаков), включая иллюстрации (1 иллюстрация форматом 190 × 270 мм = 1/6 авторского листа, или 6,7 тыс. знаков). Публикации, превышающие указанный объем, допускаются к рассмотрению только после индивидуального согласования с редакцией журнала.

Текст рукописи должен быть представлен в редколлегию в виде файла MS Word (.doc, .docx). Гарнитура Times New Roman, размер шрифта 11, межстрочный интервал 1, размеры полей – стандартные значения текстового редактора. Форматирование – выравнивание по ширине страницы, переносы слов включены, каждый новый абзац начинается с красной строки. Не допускается ручное форматирование абзацев (пробелами, лишними переводами строк, разрывами страниц).

Структура статьи

- Индекс УДК (универсальной десятичной классификации). Выравнивание по левому краю
- Название статьи. Выравнивание по центру, полужирный шрифт
- ФИО авторов (полностью). Выравнивание по центру, полужирный шрифт
- Места работы всех авторов. Выравнивание по центру, курсив
- Адреса электронной почты, ORCID авторов
- Аннотация статьи
- Ключевые слова, не более 10
- Благодарности, сведения о финансовой поддержке
- Название статьи **на английском языке**. Выравнивание по центру, полужирный шрифт
- ФИО авторов **на английском языке** (полностью). Выравнивание по центру, полужирный шрифт
- Места работы авторов **на английском язык**е. Выравнивание по центру, курсив
- Аннотация статьи **на английском языке (Abstract)**, 200–250 слов
- Ключевые слова **на английском языке (Keywords)**, не более 10
- Благодарности, сведения о финансовой поддержке **на английском язык**е, если есть соответствующий раздел на русском языке **(Acknowledgements)**
- Основной текст
- Список литературы / **References**
- Сведения об авторах

Требования к оформлению основного текста и иллюстративных материалов

Основной текст должен быть представлен в структурированном виде, рекомендуется использовать подзаголовки – например: Введение, Методика..., Выводы, Результаты, Заключение.

Подзаголовки отделяются и набираются полужирным шрифтом. В целях выделения частей текста и отдельных слов и словосочетаний допускается использование курсива или полужирного шрифта. Подчеркивание, разрядка, изменение основного кегля и выделение цветом не используются.

Иллюстрации к рукописи статьи должны быть приложены в виде отдельных файлов. При этом в тексте должно содержаться включенное изображение с указанием имени файла. Все иллюстрации, содержащие схемы, графики, алгоритмы и т. п., должны быть представлены в векторном виде (.ai, .eps, .cdr). Скриншоты и другие растровые изображения должны быть представлены в максимально высоком качестве, без каких-либо потерь и искажений (.jpg, .tif). Все иллюстрации должны иметь подрисуночную подпись – свое название. Надписи к таблицам и подписи к иллюстрациям приводятся **на двух языках (русском и английском)**.

Примеры:

Рис. 1. Диаграмма производительности...
Fig. 1. Performance diagram...

Таблица 1

Сравнение алгоритмов...

Table 1

Comparison of algorithms...

Нумерация последовательная и неразрывная от начала статьи. Не допускается использование других наименований, кроме «Рис.» / «Fig.», «Таблица» / «Table», и усложнение нумерации (например, «Рис. 3.2.»). Ссылка на иллюстрацию в тексте должна быть приведена в круглых скобках, например: (рис. 1), (табл. 1).

Формулы должны быть набраны с использованием редактора MathType либо встроенного редактора формул MS Word. Кегль основных символов – 11, греческие символы набираются прямым шрифтом, латинские – курсивом. Нумеруются только те формулы, на которые автор ссылается в тексте.

Abstract

Аннотация статьи на английском языке (Abstract) не должна быть дословным переводом русскоязычной аннотации. Раздел Abstract, как и основной текст, должен быть структурирован, в нем должно содержаться описание цели работы, методов исследования, научной значимости, выводов / результатов. Требуется качественный перевод на английский язык (при необходимости просим авторов обращаться к профессиональным переводчикам). **Объем Abstract 200–250 слов.**

Список литературы / References

Список литературы и список литературы на английском языке (References) размещаются в общем разделе. Рекомендуемое количество цитируемых в статье источников – не менее 10, в список желательно включать ссылки на актуальные работы по теме исследования, особенно в иностранных периодических изданиях.

В тексте статьи ссылки на литературу указываются цифрами в квадратных скобках, при необходимости указываются номера страниц, например: [2; 3. С. 15].

Список литературы нумеруется в порядке цитирования и оформляется в соответствии с ГОСТ Р 7.0.5-2008 на библиографическое описание (знаки тире в описании опускаются). Ссылки на неопубликованные работы, а также на Интернет-ресурсы (кроме электронных изданий, поддающихся библиографическому описанию) оформляются в виде сноски.

В Список литературы ссылки на источники следует включать на оригинальном языке опубликования. Каждый источник должен быть также оформлен на английском языке (References) по международному стандарту для публикаций в области информатики IEEE Style со следующими отличиями:

- инициалы авторов указываются после фамилии;
- название статьи не берется в кавычки, отделяется точкой;

- отсутствует союз «and» перед фамилией последнего автора;
- в диапазоне страниц – удвоенная «p» (например, «pp. 2–9»);
- год издания указывается после места издания (для книг) и сразу после названия журнала (для периодики).
- Перевод источника на английский язык:
- если источник имеет выходные данные на английском языке, то для формирования References **следует использовать именно эти данные**;
- если оригинальная публикация не содержит выходных данных на английском языке, то допускается транслитерация названия материала на латинский алфавит в сочетании с переводом на английский язык в квадратных скобках. В конце описания указывается, на каком языке написана эта работа, например, (in Russ.). При транслитерации можно воспользоваться интернет-ресурсом http://ru.translit.ru/, рекомендуется выбрать стандарт BSI. Место издания не транслитерируется, указывается полностью на английском языке, например: Moscow. Название издательства / издателя, как правило, транслитерируется. Для журналов, у которых есть официальное название на английском языке, – использовать его (проверить на сайте журнала, или, например, в библиотеке WorldCat), если названия на английском языке нет, использовать транслитерацию по системе BSI. Не следует самостоятельно переводить названия журналов.

Если у цитируемого источника есть **цифровой идентификатор DOI** (https://search. crossref. org), его требуется обязательно указывать в конце библиографической ссылки.

Примеры оформления ссылок. Каждый источник в том же пункте дублируется на английском языке (References).

Источник на русском языке, перевод на английский доступен в метаданных статьи

1. **Журавлев С. С., Рудометов С. В., Окольнишников В. В., Шакиров С. Р.** Применение модельно-ориентированного проектирования к созданию АСУ ТП опасных промышленных объектов // Вестник НГУ. Серия: Информационные технологии. 2018. Т. 16, № 4. С. 56–67. DOI 10.25205/1818-7900-2018-16-4-56-67

Zhuravlev S. S., Rudometov S. V., Okolnishnikov V. V., Shakirov S. R. Model-Based Design Approach for Development Process Control Systems of Hazardous Industrial Facilities. *Vestnik NSU. Series: Information Technologies*, 2018, vol. 16, no. 4, pp. 56–67. (in Russ.) DOI 10.25205/ 1818-7900-2018-16-4-56-67

Источник на английском языке. Оформляем согласно требованиям для References. Приводим только 1 раз.

2. **Telnov V. I.** Optimization of the Beam Crossing Angle at the ILC for E + e- and yy Collisions. *Journal of Instrumentation*, 2018, vol. 13, no. 03, pp. P03020–P03020. DOI 10.1088/1748-0221/13/03/ p03020

Метаданные источника доступны только на русском языке

3. **Жижимов О. Л., Федотов А. М., Шокин Ю. И.** Технологическая платформа массовой интеграции гетерогенных данных // Вестник НГУ. Серия: Информационные технологии. 2013. Т. 11, вып. 1. С. 24–41.

Zhizhimov O. L., Fedotov A. M., Shokin Yu. I. Tekhnologicheskaya platforma massovoi integratsii geterogennykh dannykh [Technology Platform for the Mass Integration of Heterogeneous Data]. *Vestnik NSU. Series: Information Technologies*, 2013, vol. 11, no. 1, pp. 24–41. (in Russ.)

Сведения об авторах

Последний раздел статьи – информация об авторе / авторах **на русском и английском языках**:

- ФИО полностью, ученая степень, ученое звание;
- идентификаторы автора, такие как ResearcherID (всем авторам рекомендуется использовать данные сервисы для ведения актуального списка своих публикаций);
- контактный телефон (не публикуется).

Если статья представляется на английском языке, необходимо приложить перевод на русский язык названия, аннотации, ключевых слов, сведений об авторе.

Доставка материалов

Материалы предоставляются в редакцию по электронной почте inftech@vestnik.nsu.ru.

Порядок рецензирования

Все статьи сначала проходят проверку на заимствование и только после этого отправляются на рецензирование. Редакционный совет не допускает к публикации материал, если имеется достаточно оснований полагать, что он является плагиатом.

Тип рецензирования статей – двухуровневое, одностороннее анонимное («слепое»).

Для каждой статьи редколлегией выбираются рецензенты, научная деятельность которых связана с темой представленного материала. Ответственный секретарь журнала обращается к ним с просьбой дать экспертную оценку статье либо помочь организовать рецензирование.

Рецензии для журнала «Вестник НГУ. Серия: Информационные технологии» составляются по единой схеме и подразумевают оценку по следующим критериям: соответствие тематике журнала, оригинальность и значимость результатов, качество изложения материала.

Заполненный бланк рецензии высылается на электронный адрес редакции. В зависимости от экспертных заключений статья может быть принята редакционным советом к опубликованию, рекомендована автору к доработке (с последующим повторным рецензированием либо без него) или отклонена (с предоставлением автору мотивированного отказа). Автору на электронный адрес высылается текст рецензии без указания ФИО рецензента и его контактных данных.

Все рецензии хранятся в редакции журнала не менее 5 лет. Редколлегия журнала обязуется при поступлении соответствующего запроса направлять копии рецензий в Министерство науки и высшего образования Российской Федерации.