

**НАУЧНЫЙ  
ФОРУМ**  
nauchforum.ru

ISSN 2618-9402



LXIX Студенческая международная  
заочная научно-практическая  
конференция

**ТЕХНИЧЕСКИЕ И МАТЕМАТИЧЕСКИЕ НАУКИ.  
СТУДЕНЧЕСКИЙ НАУЧНЫЙ ФОРУМ  
№2(69)**

г. МОСКВА, 2024



# ТЕХНИЧЕСКИЕ И МАТЕМАТИЧЕСКИЕ НАУКИ. СТУДЕНЧЕСКИЙ НАУЧНЫЙ ФОРУМ

*Электронный сборник статей по материалам LXIX студенческой  
международной научно-практической конференции*

№ 2 (69)  
Февраль 2024 г.

Издается с февраля 2018 года

Москва  
2024

УДК 62+51  
ББК 30+22.1  
Т38

Председатель редколлегии:

**Лебедева Надежда Анатольевна** – доктор философии в области культурологии, профессор философии Международной кадровой академии, г. Киев, член Евразийской Академии Телевидения и Радио.

Редакционная коллегия:

**Волков Владимир Петрович** – кандидат медицинских наук, рецензент АНС «СибАК»;

**Елисеев Дмитрий Викторович** – кандидат технических наук, доцент, начальник методологического отдела ООО "Лаборатория институционального проектного инжиниринга";

**Захаров Роман Иванович** – кандидат медицинских наук, врач психотерапевт высшей категории, кафедра психотерапии и сексологии Российской медицинской академии последипломного образования (РМАПО) г. Москва;

**Зеленская Татьяна Евгеньевна** – кандидат физико-математических наук, доцент, кафедра высшей математики в Югорском государственном университете;

**Карпенко Татьяна Михайловна** – кандидат философских наук, рецензент АНС «СибАК»;

**Костылева Светлана Юрьевна** – кандидат экономических наук, кандидат филологических наук, доц. Российской академии народного хозяйства и государственной службы при Президенте РФ (РАНХиГС), г. Москва;

**Попова Наталья Николаевна** – кандидат психологических наук, доцент кафедры коррекционной педагогики и психологии института детства НГПУ;

**Т38 Технические и математические науки. Студенческий научный форум.** Электронный сборник статей по материалам LXIX студенческой международной научно-практической конференции. – Москва: Изд. «МЦНО». – 2024. – № 2 (69) / [Электронный ресурс] – Режим доступа. – URL: [https://nauchforum.ru/archive/SNF\\_tech/2\(69\).pdf](https://nauchforum.ru/archive/SNF_tech/2(69).pdf)

Электронный сборник статей LXIX студенческой международной научно-практической конференции «Технические и математические науки. Студенческий научный форум» отражает результаты научных исследований, проведенных представителями различных школ и направлений современной науки.

Данное издание будет полезно магистрам, студентам, исследователям и всем интересующимся актуальным состоянием и тенденциями развития современной науки.

## **Оглавление**

<b>Секция 1. Технические науки</b>	<b>4</b>
СРАВНЕНИЕ АЛГОРИТМОВ СОРТИРОВКИ: АНАЛИЗ ЭФФЕКТИВНОСТИ Блинов Роман Викторович Бычков Кирилл Вячеславович Кирчева Алина Сергеевна Мамедов Илькин Вахид оглы	4
ЕРС-КОНТРАКТЫ В РОССИИ Козлов Константин Владимирович	12
УЛУЧШЕНИЕ СКРЕБКОВОГО КОНВЕЙЕРА: ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ И НАДЕЖНОСТИ Тунгат Анар	17
АНАЛИЗ СЕМАНТИЧЕСКОЙ БЛИЗОСТИ ПРИ ОПРЕДЕЛЕНИИ ЖАНРОВОЙ ПРИНАДЛЕЖНОСТИ Юрков Андрей Александрович Ивайловский Евгений Олегович	28

# СЕКЦИЯ 1.

## ТЕХНИЧЕСКИЕ НАУКИ

### СРАВНЕНИЕ АЛГОРИТМОВ СОРТИРОВКИ: АНАЛИЗ ЭФФЕКТИВНОСТИ

***Блинов Роман Викторович***

*студент,  
Сибирский государственный индустриальный университет,  
РФ, г. Новокузнецк*

***Бычков Кирилл Вячеславович***

*студент,  
Сибирский государственный индустриальный университет,  
РФ, г. Новокузнецк*

***Кирчева Алина Сергеевна***

*студент,  
Сибирский государственный индустриальный университет,  
РФ, г. Новокузнецк*

***Мамедов Илькин Вахид оглы***

*студент,  
Сибирский государственный индустриальный университет,  
РФ, г. Новокузнецк*

Сортировка является одной из фундаментальных операций в области компьютерных алгоритмов. Она позволяет упорядочить набор данных в заданном порядке и является неотъемлемой частью множества приложений, начиная от базовых операций с данными до сложных алгоритмов машинного обучения и анализа данных.

Существует множество алгоритмов сортировки, каждый из которых имеет свои особенности и характеристики. Одной из важных задач при выборе алгоритма сортировки является анализ и сравнение их эффективности. Понимание производительности алгоритмов сортировки позволяет разработчикам искусно выбирать наиболее подходящий алгоритм для конкретных ситуаций, учитывая ограничения по времени и памяти.

Мы рассмотрим некоторые из наиболее часто используемых методов сортировки, применяя Python как язык программирования для иллюстрации:

1. Сортировка пузырьком (Bubble Sort) – это алгоритм сортировки, который сравнивает два соседних элемента массива и меняет их местами, пока они не будут в нужном порядке, имеет квадратичную сложность  $O(n^2)$  в среднем и худшем случаях и  $O(n)$  в лучшем случае, когда массив уже отсортирован [1]. Обычно используется для сортировки небольших массивов.

Алгоритм сортировки пузырьком работает следующим образом:

- a. Начинаем с первого элемента массива;
- b. Сравниваем первый элемент с вторым. Если первый элемент больше второго, меняем их местами;
- c. Переходим к следующему элементу массива. Снова сравниваем его со следующим и меняем местами, если он больше;
- d. Продолжаем этот процесс до конца массива. Таким образом, самый большой элемент перемещается на последнее место в первой итерации;
- e. Повторяем те же шаги для оставшихся элементов массива, исключая уже отсортированные. После каждой итерации очередной наибольший элемент становится на свое место;
- f. Алгоритм заканчивается, когда все элементы массива отсортированы.

Пример сортировки пузырьком на языке Python представлен на рисунке 1

```
def bubbleSort(array) -> list:
    length = len(array)
    for i in range(length):
        for j in range(length - i - 1):
            if array[j] > array[j + 1]:
                temp = array[j]
                array[j] = array[j + 1]
                array[j + 1] = temp
    return array
```

*Рисунок 1. Сортировка пузырьком на языке Python*

2. Сортировка вставками (Insertion Sort) – это алгоритм сортировки, который постепенно строит отсортированную последовательность, проходя по списку и вставляя каждый элемент в правильное место. Он обладает линейной сложностью  $O(n)$  в лучшем случае (когда список уже отсортирован) и квадратичной сложностью  $O(n^2)$  в среднем и худшем случаях [2].

- a. Алгоритм сортировки вставками работает следующим образом;
- b. Считаем первый элемент массива отсортированным;
- c. Берем следующий элемент и сохраняем его в отдельной переменной (назовем ее ключ);
- d. Сравниваем ключ с элементами в отсортированной части массива, начиная с последнего. Если ключ меньше текущего элемента, то сдвигаем текущий элемент на одну позицию вправо. Повторяем этот процесс, пока не найдем элемент, который меньше или равен ключу, или не дойдем до начала массива;
- e. Вставляем ключ на найденное место или в начало массива, если он меньше всех элементов в отсортированной части;
- f. Повторяем шаги 2-4 для оставшихся элементов в неотсортированной части массива, пока не отсортируем весь массив.

Пример сортировки вставками на языке Python представлен на рисунке 2

```
def insertionSort(array) -> list:
    for i in range(1, len(array)):
        key = array[i]
        j = i - 1
        while j >= 0 and array[j] > key:
            array[j + 1] = array[j]
            j = j - 1
        array[j + 1] = key
    return array
```

**Рисунок 2. Сортировка вставками на языке Python**

3. Сортировка выбором (Selection Sort): этот алгоритм сортировки на каждом шаге находит минимальный элемент из неотсортированной части списка и меняет его местами с первым неотсортированным элементом. Сложность



алгоритма сортировки выбором составляет  $O(n^2)$  в лучшем, среднем и худшем случаях, так как на каждой итерации требуется пройти по всему неотсортированному массиву [1]. Этот алгоритм не подходит для больших объемов данных, так как он работает очень медленно. Однако он прост в понимании и реализации и не требует дополнительной памяти.

Алгоритм сортировки выбором работает следующим образом:

- a. Считаем первый элемент массива наименьшим;
- b. Сравниваем наименьший элемент с остальными элементами массива. Если находим элемент меньше наименьшего, запоминаем его как новый наименьший;
- c. После прохода по всему массиву меняем местами наименьший элемент с первым элементом неотсортированной части массива. Таким образом, наименьший элемент оказывается на первом месте отсортированной части массива;
- d. Повторяем шаги 1-3 для оставшихся элементов неотсортированной части массива, пока не отсортируем весь массив.

Пример сортировки выбором на языке Python представлен на рисунке 3

```
def selectionSort(array) -> list:
    length = len(array)
    for i in range(0, length - 1):
        minIndex = i
        for j in range(i + 1, length):
            if array[j] < array[minIndex]:
                minIndex = j
        temp = array[i]
        array[i] = array[minIndex]
        array[minIndex] = temp
    return array
```

*Рисунок 3. Сортировка выбором на языке Python*

4. Сортировка слиянием (Merge Sort): это эффективный алгоритм сортировки, который основан на принципе "разделяй и властвуй". Он разбивает список на две половины, рекурсивно сортирует каждую половину, а затем объединяет их, чтобы получить отсортированный список. Сортировка слиянием имеет сложность



$O(n \log n)$  во всех случаях [3]. Этот алгоритм эффективен для больших объемов данных, но требует дополнительной памяти для хранения временных массивов.

Алгоритм сортировки слиянием работает следующим образом:

а. Если массив состоит из одного элемента, то он уже отсортирован и возвращается как результат;

б. Если массив состоит из более чем одного элемента, то он делится на две равные или почти равные части;

с. Каждая часть сортируется рекурсивно с помощью сортировки слиянием;

д. Две отсортированные части объединяются в один отсортированный массив с помощью процедуры слияния, которая сравнивает элементы двух частей и копирует меньший элемент в результирующий массив, пока обе части не будут исчерпаны.

Пример сортировки слиянием на языке Python представлен на рисунке 4

```
def mergeSort(array) -> list:
    if len(array) > 1:
        mid = len(array)//2
        left = array[:mid]
        right = array[mid:]
        mergeSort(left)
        mergeSort(right)
        i = j = k = 0
        while i < len(left) and j < len(right):
            if left[i] <= right[j]:
                array[k] = left[i]
                i += 1
            else:
                array[k] = right[j]
                j += 1
            k += 1
        while i < len(left):
            array[k] = left[i]
            i += 1
            k += 1
        while j < len(right):
            array[k] = right[j]
            j += 1
            k += 1
    return array
```

*Рисунок 4. Сортировка слиянием на языке Python*

5. Быстрая сортировка (Quick Sort): это один из самых быстрых алгоритмов сортировки в среднем случае. Он также использует принцип "разделяй и властвуй", выбирая опорный элемент и разделяя список на две части: элементы, меньшие опорного, и элементы, большие опорного. Затем он рекурсивно сортирует каждую часть. Быстрая сортировка в среднем случае имеет сложность  $O(n \log n)$ , но в худшем случае может иметь квадратичную сложность  $O(n^2)$  [4].

Быстрая сортировка работает следующим образом:

а. Если массив состоит из одного или нуля элементов, то он уже отсортирован и возвращается как результат;

б. Если массив состоит из более чем одного элемента, то выбирается один элемент в качестве опорного. Это может быть любой элемент массива, но от выбора опорного зависит эффективность алгоритма. Существуют разные способы выбора опорного элемента, например, первый, последний, средний, случайный или медианный элемент массива;

с. Переставить элементы массива так, чтобы все элементы, меньшие опорного, оказались слева от него, а все элементы, большие или равные опорному, оказались справа от него. Это называется разбиением массива. Существуют разные способы выполнения разбиения, например, схема Ломута, схема Хоара, схема Тарьяна и другие;

д. Рекурсивно применить быструю сортировку к левой и правой части массива, не включая опорный элемент;

е. Объединить отсортированные левую и правую части массива с опорным элементом в один отсортированный массив.

Пример быстрой сортировки используя схему разбиения Хоара на языке Python представлен на рисунке 5

```

def quickSort(array, start, end) -> list:
    if end - start > 1:
        part = partition(array, start, end)
        quickSort(array, start, part)
        quickSort(array, part + 1, end)
    return array

def partition(array, start, end):
    pivot = array[start]
    i = start + 1
    j = end - 1
    while True:
        while (i <= j and array[i] <= pivot):
            i = i + 1
        while (i <= j and array[j] >= pivot):
            j = j - 1
        if i <= j:
            temp = array[i]
            array[i] = array[j]
            array[j] = temp
        else:
            temp = array[start]
            array[start] = array[j]
            array[j] = temp
    return j

```

**Рисунок 5 - быстрая сортировка используя схему разбиения Хоара на языке Python**

Для сравнения различных методов сортировки в Python мы воспользуемся библиотекой `time`. Эта библиотека позволяет замерять время выполнения кода, что полезно при оценке производительности алгоритмов.

У нас есть список из 441 числа, который мы будем сортировать. Для каждого метода сортировки выполним десять итераций и замерим среднее время выполнения, которое будет представлено в таблице 1.

**Таблица 1.**

**Сравнение различных методов сортировки**

Название метода	Среднее время выполнения, секунд
Сортировка пузырьком	0.005481553077697754
Сортировка вставками	0.002293300628662109
Сортировка выбором	0.0025913238525390623
Сортировка слиянием	0.000506448745727539
Быстрая сортировка	0.0002918243408203125

Таким образом, при выборе метода сортировки рекомендуется использовать быструю сортировку или сортировку слиянием для достижения наилучшей производительности.

### **Список литературы:**

1. Hammad J. A comparative study between various sorting algorithms // International Journal of Computer Science and Network Security (IJCSNS). – 2015. – Т. 15. – №. 3. – С. 11.
2. Bender M.A., Farach-Colton M., Mosteiro M.A. Insertion sort is  $O(n \log n)$  // Theory of Computing systems. – 2006. – Т. 39. – С. 391-397.
3. Lobo J., Kuwelkar S. Performance analysis of merge sort algorithms //2020 International Conference on Electronics and Sustainable Communication Systems (ICESC). – IEEE, 2020. – С. 110-115.
4. Rajput I.S., Kumar B., Singh T. Performance comparison of sequential quick sort and parallel quick sort algorithms //International Journal of Computer Applications. – 2012. – Т. 57. – №. 9. – С. 14-22.

## **ЕРС-КОНТРАКТЫ В РОССИИ**

***Козлов Константин Владимирович***

*Федеральное Государственное Бюджетное  
Образовательное Учреждение  
высшего образования МИРЭА –  
Российский технологический университет,  
РФ, г. Москва*

## **EPC-CONTRACTS IN RUSSIA**

***Konstantin Kozlov***

*Federal State Budgetary Educational Institution  
of Higher Education MIREA - Russian Technological University,  
Russia, Moscow*

**Аннотация.** В данной статье подчеркивается увеличение объема проектов ЕРС в России, особенно в нефтегазовом секторе, и важность эффективного управления качеством в этих проектах. ЕРС-контракт, включающий инжиниринг, закупку и строительство, является комплексным контрактом, который возлагает ответственность за весь цикл проекта на Подрядчика. Описываются этапы инжиниринга, закупки и строительства, акцентируя внимание на важности каждого этапа и потенциальных рисках и вызовах, связанных с ними. В статье подчеркивается значимость этапа закупок в проектах ЕРС, а также риски и потенциальные проблемы, связанные с ними.

**Abstract.** This paper emphasizes the increasing volume of EPC projects in Russia, especially in the oil and gas sector, and the importance of effective quality management in these projects. An EPC contract including engineering, procurement and construction is a complex contract that places responsibility for the entire project cycle on the Contractor. The engineering, procurement and construction phases are described, emphasizing the importance of each phase and the potential risks and challenges associated with them. The article emphasizes the importance of the procurement phase in EPC projects and the risks and potential challenges associated with them.

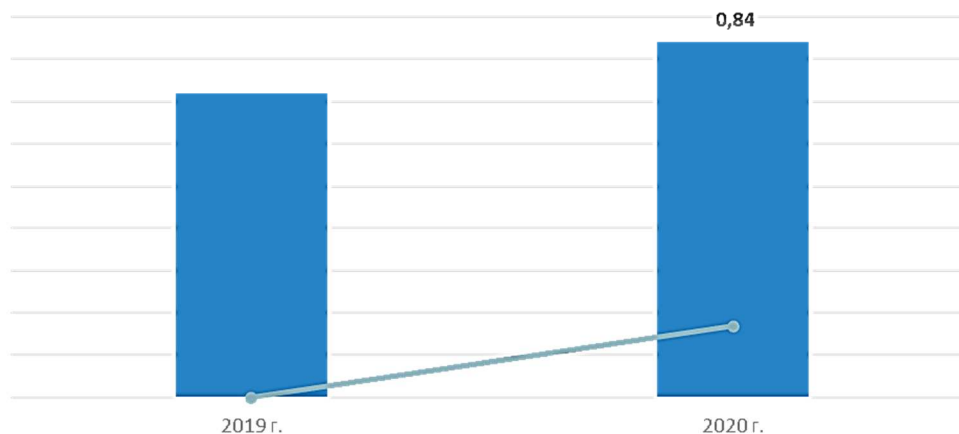
**Ключевые слова:** EPC-контракты в России, управление качеством, инжиниринг, закупка, строительство, риски, вызовы реализации проектов на этапе закупок.

**Keywords:** EPC projects in Russia, quality management, engineering, procurement, construction, risks, challenges of project implementation at the procurement stage.

### **Увеличение объемов реализации строительных и инженерных проектов по EPC-контрактам в России**

Согласно анализу динамики рынка использование EPC-контрактов в СПГ и нефтегазохимической отраслях будет составлять не менее 50% от общего числа проектов в ближайшие годы. Переход на использование EPC-контрактов может упростить и ускорить процесс реализации государственных проектов.

**Указан недопустимый источник.**



**Рисунок 1. Объем рынка EPC-контрактов в России в стоимостном выражении в 2019-2020 гг., трлн. руб. и % прироста** Указан недопустимый источник.

В 2022-2023 годах зарубежные EPC-контракторы вышли из российского рынка, что привело к проблемам с поставками оборудования, заключенными до введения санкций. Невозможность получить необходимое оборудование замед-

лила или приостановила работы и потребовала внесения изменений в проект.

**Указан недопустимый источник.**

Процентное соотношение участия отечественных ЕРС-контракторов в инвестиционных проектах в России варьировалось от 20% до 25% (за тот же период). Эксперты прогнозируют быстрый рост рынка ЕРС-контрактов в России и ожидают увеличения доли отечественных компаний в этой сфере. Однако рыночные участники и эксперты считают, что отечественным компаниям не хватает опыта и навыков, а также импортных технологий.

Важно отметить, что использование подобных контрактов в России началось в конце XX века, но развитие данной сферы до сих пор оставляет желать лучшего. В России требуются комплексные изменения в регулировании данного вида контрактов в рамках законодательства. **Указан недопустимый источник.**

Распоряжением Правительства Российской Федерации от 11 июня 2020 г. N 1546-р утвержден план мероприятий ("дорожная карта") в области инжиниринга и промышленного дизайна, включающая сокращение отставания Российской Федерации от развитых стран в части ЕРС(М)-компетенций для выполнения высокотехнологичных проектов в отраслях металлургии, химии и иных отраслях обрабатывающей промышленности, освоения месторождений шельфа и Восточной Сибири, создания новых отечественных технологий в области сжижения природного газа. **Указан недопустимый источник.**

### **ЕРС-контракт**

ЕРС-контракт – договор проектирования, поставки оборудования и строительства, заключаемый между заказчиком и подрядчиком. Согласно этому договору, подрядчик выполняет полный цикл работ и ответственен за риски, связанные со строительством.

Расшифровка аббревиатуры ЕРС в названии контракта включает:

Е – Engineering (инжиниринг);

Р – Procurement (закупка);

С – Construction (строительство).



Для успешной реализации проекта по контракту ЕРС, заказчик и ЕРС-подрядчик должны выполнять следующие этапы:

1. Инжиниринг (Engineering): на этом этапе проводится проектирование проекта, разработка технических решений и технический анализ. Заказчик определяет требования к проекту, а ЕРС-подрядчик разрабатывает техническое задание, проводит исследования и разрабатывает проектные решения.

2. Закупка (Procurement): на этом этапе осуществляется закупка оборудования, материалов и услуг для реализации проекта. ЕРС-подрядчик осуществляет поиск поставщиков, проводит предварительный отбор, проводит торги и заключает контракты на поставку необходимых материалов и оборудования.

3. Строительство (Construction): на этом этапе происходит физическое выполнение строительных работ и установка оборудования. ЕРС-подрядчик организует строительные работы, координирует подрядчиков, контролирует сроки и качество выполнения работ.

### **Важность обеспечения и контроля качества на этапе закупок (P)**

Этап закупок (P) в проектах по ЕРС-контрактам имеет большое значение, поскольку качество материалов и оборудования влияет на успешность реализации проекта. Задержки в поставках оборудования могут привести к задержкам в проекте, поэтому контроль на этом этапе является особенно важным.

Однако, уровень контроля зачастую ниже на этапе закупок по сравнению с другими этапами проекта, особенно если используется аутсорсинг и субподряд. Это может означать, что качество закупаемого оборудования и материалов может быть недостаточно высоким, что влечет за собой потенциальные риски и вызовы для проекта. Исследования показывают, что задержки в поставках оборудования, особенно тех, которые требуют длительного изготовления, являются одним из наиболее значимых факторов риска для ЕРС-проектов. **Указан недопустимый источник.** Вместе с тем, уровень контроля зачастую ниже по сравнению с проектированием и строительством. **Указан недопустимый источник.** Поэтому и контроль, и обеспечение качества на этапе закупки и поставки оборудования являются важными задачами.

Дальнейшие исследования в этой области могут помочь разработать методологии и подходы, которые позволят обеспечить более высокий уровень контроля качества на этапе закупок и предотвратить возможные задержки в проекте. Это может включать в себя разработку более эффективных систем контроля качества, а также улучшение коммуникации и сотрудничества между различными сторонами проекта, включая поставщиков оборудования и материалов.

### Список литературы:

1. Инвестиционная Группа «Профессиональные Комплексные Решения» (ПКР). Рынок EPC-контрактов и контрактов генерального подряда в РФ. [В Интернете] <https://prcs.ru/analytics-article/rynok-epc-kontraktov-i-kontraktov-generalnogo-podryada/>.
2. Агентство DISCOVERY Research Group. Анализ рынка EPC-контрактов в России. [В Интернете] <https://drgroup.ru/Analiz-rynka-EPC-kontraktov-v-Rossii.html>.
3. ООО «Компания Р-Медиа». EPC-контракторы – уйти нельзя остаться. Сетевое издание РУБЕЖ. [В Интернете] 26 сентября 2023 г. <https://rubezh.ru/infografika/epc-kontraktory-uyti-nelzya-ostatsya>.
4. Глебова А.Г., Васильев И.А. Применение мирового опыта EPCM-контрактов в России. Вестник Забайкальского государственного университета. 2022 г., Т. 28, 4, стр. 86-95.
5. Распоряжение Правительства РФ от 11 июня 2020 г. N 1546-р. Текст распоряжения опубликован на "Официальном интернет-портале правовой информации" ([www.pravo.gov.ru](http://www.pravo.gov.ru)) 15 июня 2020 г., в Собрании законодательства Российской Федерации от 22 июня 2020 г. N 25 ст. 3934.
6. Khasani, J.U.D. Hatmoko and R.R. Mapping Delay Risks of EPC Projects: A Case Study of A Platform and Subsea Pipeline of An Oil and Gas Project. [В Интернете] 2019 г. <https://iopscience.iop.org/article/10.1088/1757-899X/598/1/012095>.
7. K.T Yeo, J.H Ning,. Integrating supply chain and critical chain concepts in engineer-procure-construct (EPC) projects. International Journal of Project Management, Volume 20, Issue 4,. [В Интернете] Pages 253-262,. <https://www.sciencedirect.com/science/article/pii/S0263786301000217>.

## **УЛУЧШЕНИЕ СКРЕБКОВОГО КОНВЕЙЕРА: ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ И НАДЕЖНОСТИ**

*Тунгат Анар*

*магистрант,*

*Казахский агротехнический исследовательский университет,*

*Казахстан, г. Астана*

Один из наиболее инновационных средств транспортировки, обеспечивающих высокую производительность и техно-экономическую эффективность при обработке значительных объемов грузов, представляет собой конвейер. В современных условиях массового и серийного производства конвейеры становятся неотъемлемой частью технологического процесса, регулируя темп производства, обеспечивая его ритмичность, способствуя увеличению производительности труда и расширению выпуска продукции. Они эффективно решают задачи комплексной механизации и автоматизации транспортно-технологических процессов.

Широкое внедрение конвейеров происходит в различных сферах национального хозяйства, где их тяговые и грузоперевозочные элементы применяются в роли транспортных и подающих узлов для строительной-дорожной техники, сельскохозяйственных машин, полиграфического оборудования и других технических средств.

Особенно перспективным становится использование конвейеров в гибких автоматизированных системах транспортировки и хранения на транспортно-перегрузочных комплексах. В промышленности конвейеры применяются в качестве высокопроизводительных транспортных средств, перемещающих грузы как внутри завода, так и, в некоторых случаях, на внешнем транспорте.

Они выполняют роль транспортных агрегатов для мощных перегрузочных устройств, таких как мостовые перегружатели, отвалообразователи и прочие [2, с.57].

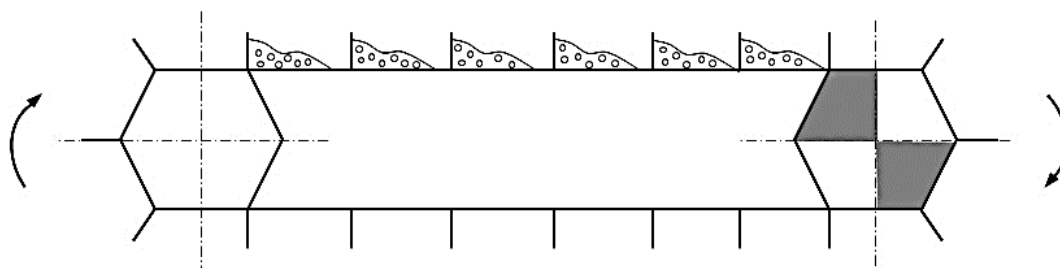
Кроме того, конвейеры используются для перемещения грузов и изделий по технологическому процессу поточного производства, обеспечивая регулирование

темпа производства и объединяя, в некоторых случаях, функции накопителей и распределителей грузов и изделий по отдельным технологическим линиям. Машины и передаточные устройства в технологических автоматических линиях изготовления и обработки деталей и узлов изделий также успешно используют конвейеры.

Наиболее часто встречаемыми видами конвейеров являются ленточные и скребковые. Скребковые конвейеры привлекают внимание своей актуальностью и эффективностью при транспортировке абразивных, грубых или клейких материалов, что может представлять сложности для других типов конвейеров. Они предоставляют возможность перевозки материалов на большие расстояния с минимальными потерями или повреждениями, обеспечивая при этом наилучшую сохранность груза. Изучение и понимание характеристик эксплуатации скребковых конвейеров позволяет оптимизировать их функционирование, добиваясь максимальной эффективности в различных отраслях промышленности [3, с.123]. Скребковые конвейеры представляют собой тип конвейеров, оснащенных скребками, прикрепленными к ленте или цепи. Скребки выполняют роль опоры и транспортировки материалов, преодолевая абразивные, грубые или клейкие свойства материалов. Горизонтальные или наклонные скребковые конвейеры могут применяться в зависимости от угла наклона, требуемого для конкретных задач. Одним из ключевых отличий является отсутствие прецизионных роликов, что делает их подходящими для условий, требующих высокой прочности и надежности.

Согласно условиям эксплуатации и характеру работы, скребковые конвейеры классифицируются как стационарные и переносные. В то время как стационарные конвейеры широко применяются как транспортные машины, переносные конвейеры находят преимущественное применение в роли промежуточного транспорта в очистных выработках. Оптимальные размеры и конструкция конвейера зависят от различных факторов, таких как свойства перемещаемого материала, соотношение размеров скребков, шаг скребков, угол наклона конвейера и скорость движения скребков. Все эти факторы оказывают влияние на эффективность

транспортировки материала и зависят от конкретных условий применения конвейера и характера перемещаемого груза [1, с.117].



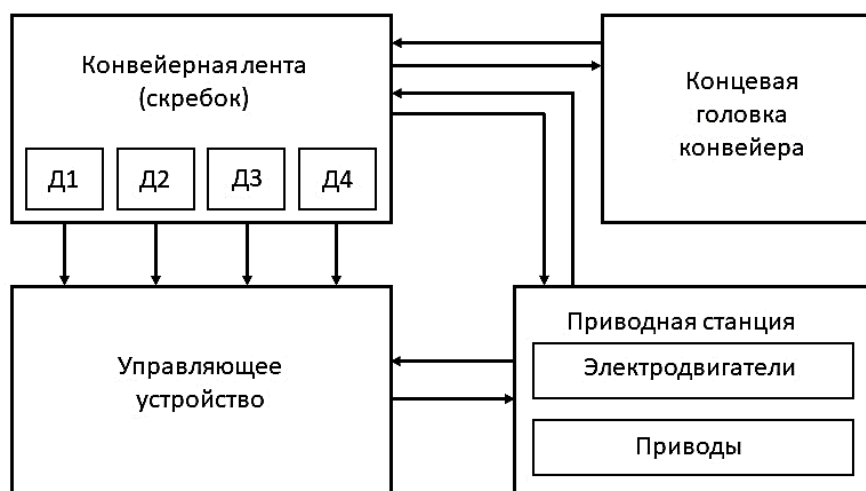
**Рисунок 1. Схема рабочих органов скребкового конвейера**

При осуществлении грузоперевозок на скребковом конвейере загрузка груза осуществляется через отверстие 2, а затем транспортировка происходит по желобу или трубе при помощи скребков (рис. 1). Непосредственную работу выполняет нижняя часть конвейера, где скребки 4 прочно закреплены на тяговой цепи 3 и перемещают груз внутри желоба 5. Тяговая цепь с прикрепленными скребками проходит через приводную (6) и натяжную (1) звездочки. Сечение скребков может быть различным: прямоугольное, полукруглое или трапециевидное. Самыми частыми материалами для изготовления скребков являются штампованный листовой металл и литые материалы, в то время как желобы чаще выполнены из металла, иногда используется дерево [7, с. 91]. Тенденция развития скребковых конвейеров направлена на увеличение их мощности, протяженности, эффективности и надежности. Существует ряд преимуществ у скребковых конвейеров, таких как возможность гибкой настройки длины конвейера, легкость интеграции в конструкцию крепежа, способность использования комбайна в роли опоры и высокая производительность. Впрочем, в процессе эксплуатации возникают определенные трудности, такие как запуск загруженных конвейеров и недостаточная надежность электроприводов при высоких нагрузках. Отмечается также значительное энергопотребление скребковых конвейеров для преодоления сопротивления.

Основные поломки, связанные с неисправностью тягового механизма, требуют детального исследования режимов запуска и торможения цепи, так как именно в эти моменты возникают высокие динамические нагрузки, влияющие на прочность и надежность всей системы [10, с. 312]. Схема управления скребковым конвейером включает в себя несколько основных блоков:

- Управляющее устройство (программируемый контроллер) – исполнитель алгоритмов управления и контроля процесса транспортировки.
- Конвейерная лента – цепь, соединяющая скребки с приводом, и сами скребки, передающие механическое воздействие на материал во время транспортировки.
- Датчики контроля процесса транспортировки: Д1 – датчик скорости рабочего органа конвейера, Д2 – датчик температуры, Д3 – датчик заполнения полостей и зазоров, Д4 – датчик обрыва цепи.
- Приводная станция – сочетание электрического двигателя и привода, передающего вращение.
- Концевая головка – передача усилия на основной элемент конструкции.

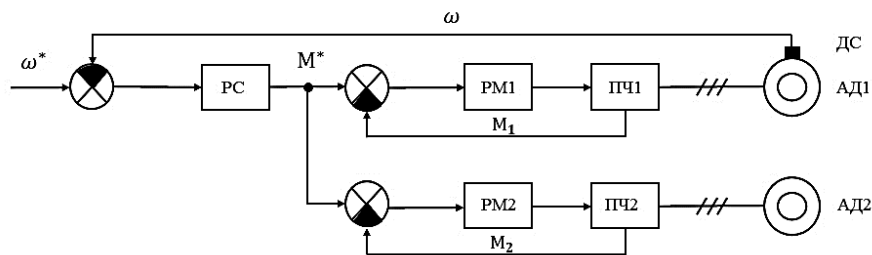
Развитие технологии скребковых конвейеров направлено на преодоление этих проблем и повышение их эффективности в промышленности [5, с. 56].



*Рисунок 2. Блок-схема управления скребкового конвейера*

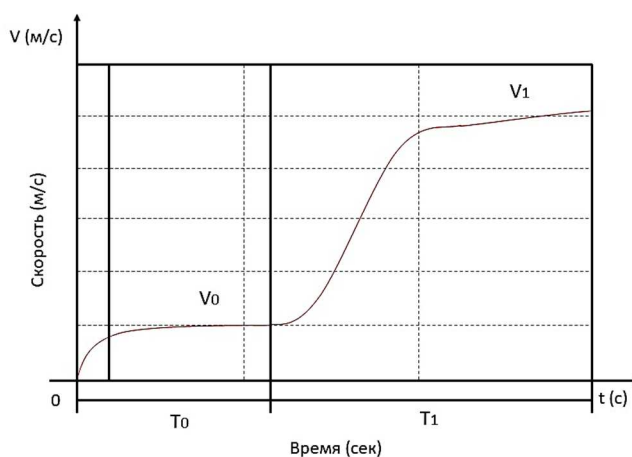
На сегодняшний день основным направлением увеличения эффективности и надежности управления скребковым конвейером становится возможность воздействия на пусковые режимы с использованием электродвигателей головного и хвостового приводов. В этом контексте активно исследуются алгоритмы пуска и остановки конвейерной ленты с целью разработки более эффективных методов управления. Основные динамические характеристики скребкового конвейера, такие как натяжение, скорость, смещение, ускорение и общее сопротивление движению, подвержены воздействию различных внешних факторов. Сюда включаются движущая сила, тормозное усилие, эффект многоугольника, смещение с автоматической регулировкой системы натяжения, изменения объема породы на конвейере и другие факторы, оказывающие непосредственное воздействие на нагрузки [4, с. 82]. Основная задача управления динамикой скребкового конвейера заключается в контроле процесса запуска установки в соответствии с разницей между установленными параметрами и фактическими показателями. При этом требуется обеспечить, чтобы выходные параметры соответствовали установленным требованиям. В данной работе акцент сделан на разработке оптимальных алгоритмов для двухступенчатого пуска скребкового конвейера. Это в конечном итоге сводится к поиску наилучших характеристик выходного сигнала на приводные установки. Рассматривается типичный двухприводный скребковый конвейер, приведенный на примере модели "Анжера-30". В данном случае использовались асинхронные взрывозащищенные двигатели DAMEL SG3 450 4/8 мощностью 200 кВт. Каждый из приводов (головной и хвостовой) оборудован собственным преобразователем частоты (ПЧ). Головной двигатель обладает регулятором скорости (РС) и контуром регулирования крутящего момента (PM1), в то время как хвостовой двигатель имеет только контур регулирования крутящего момента (PM2). Такая конфигурация обеспечивает равномерное распределение нагрузки между головным и хвостовым приводами, способствуя оптимизации работы всей системы [11, с. 75].





**Рисунок 3. Структура привода скребкового конвейера**

Основная идея двухступенчатого пуска заключается в последовательном воздействии на работу двигателя. Сначала происходит предварительное намагничивание двигателя после чего осуществляется пуск на низкую скорость, которая обычно составляет 10% от номинальной, и продолжается в течение определенного времени [8, с. 34]. Затем следует переход на полную скорость, длительность которого составляет 2 секунды. Этот временной интервал необходим для завершения всех переходных процессов в тяговых органах. Схематичное представление данного процесса проиллюстрировано на рисунке 4.



**Рисунок 4. Управляемая стартовая оптимизированная кривая**

На данной иллюстрации демонстрируется, что начальная скорость медленного пуска на первом уровне ( $V_0$ ) равна 0.1 умножить на скорость полного пуска ( $V_1$ ), и при этом выполняется условие, что произведение  $V_0 \times V_0$  должно быть больше или равно минимальному пути ( $\Delta l$ ), пройденному скребковым

конвейером. Здесь  $V_0$  представляет собой начальную скорость при медленном пуске,  $T_0$  – время медленного пуска при скорости  $V_0$ ,  $\Delta l$  – минимальное расстояние, которое конвейер должен преодолеть и которое больше или равно упругому удлинению цепи [12, с. 15]. Из-за сложности математической модели скребкового конвейера оптимизация процесса пуска рассматривается с использованием теории вязкоупругой динамики. Предполагается, что перед полным пуском конвейер предварительно запускается чередой микроперемещений. В момент запуска направление силы трения остается постоянным, а цепь рассматривается как линейное вязкоупругое тело. Модель Фойгта используется для описания сегмента соединения звеньев цепи. После этого увеличение скорости происходит по форме "S" [9, с. 89].

$$V(t) = \begin{cases} \frac{2V_e}{T^2} t^2, & 0 \leq t \leq \frac{T}{2} \\ V_e \left( -1 + \frac{4t}{T} - \frac{2t^2}{T^2} \right), & \frac{T}{2} \leq t \leq T \end{cases} \quad (1)$$

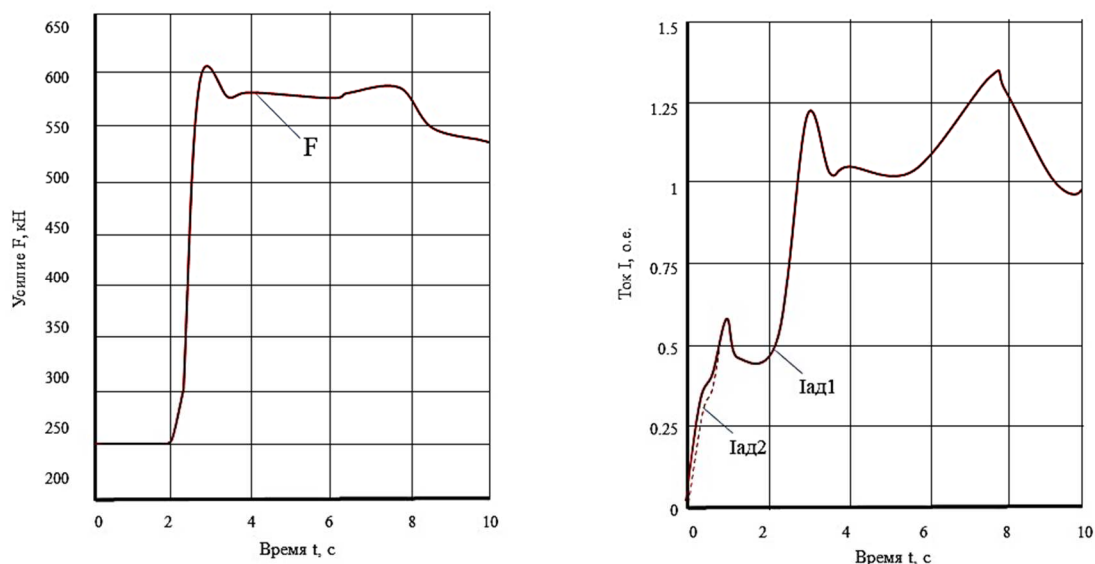
где  $V_e = V_{\text{НОМ}} - V_0$ , скорость второй «ступени» м/с;  $T$ - время увеличения скорости с  $V_0$  до  $V_{\text{НОМ}}$ , с.

При этом ускорение изменяется по треугольной форме:

$$a(t) = \begin{cases} \frac{4V_e}{T^2} t, & 0 \leq t \leq \frac{T}{2} \\ \frac{4V_e}{T} \left( 1 - \frac{t}{T} \right), & \frac{T}{2} \leq t \leq T \end{cases} \quad (2)$$

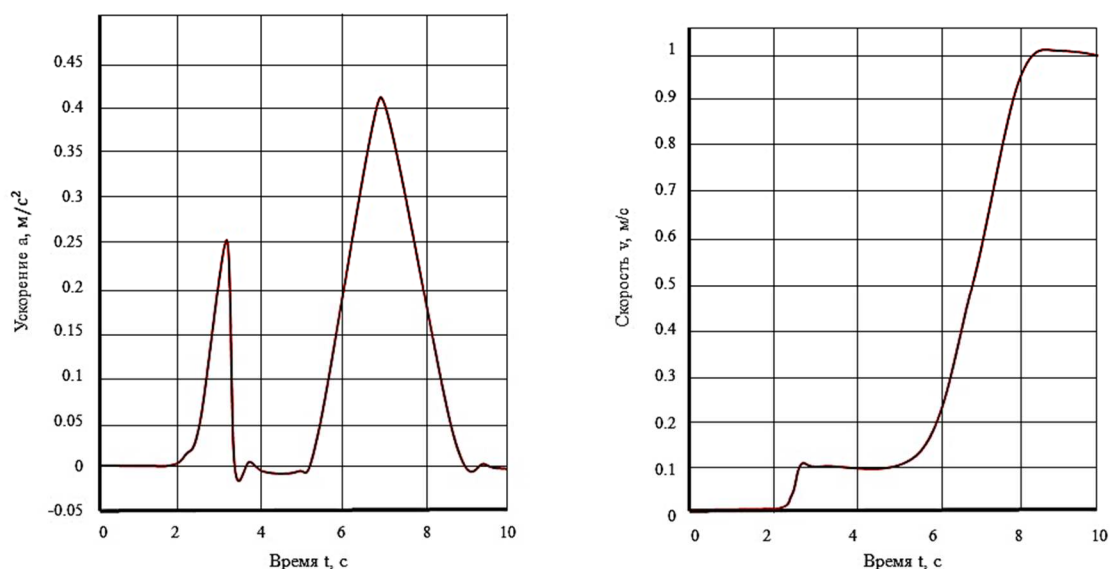
С целью более наглядного представления исследовательского процесса данная работа воспользуется графиками, полученными при математическом моделировании через программный пакет MATLAB. Использование этого инструмента позволяет провести более точные и систематические исследования. На рисунке 5 представлены зависимости изменений токов головного ( $I_{ад1}$ ) и хвостового ( $I_{ад2}$ ) электродвигателей, а также усилия натяжения цепи ( $F$ ) у головной приводной звездочки при двухступенчатом пуске при полной загрузке

конвейера. Анализ графиков показывает, что максимальное значение тока составляет 1,35 от базового, что оказывается ниже перегрузочной способности преобразователя частоты (ПЧ) [6, с. 211]. Максимальное усилие в цепи достигает 600 кН, что также находится в пределах допустимого значения с учетом сохранения запаса прочности.

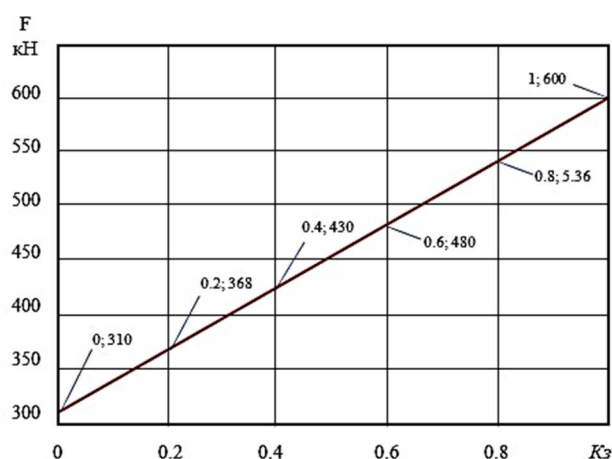


**Рисунок 5. Кривые изменения усилия в цепи (а) и тока двигателя (б)**

На изображении 5 представлен график изменения ускорения (а) и скорости (б) грузовой ветви конвейера. На данной диаграмме наглядно прослеживаются ключевые этапы двухступенчатого пуска: с 2-й по 3-ю секунду - разгон и установление низкой (маневровой) скорости, а с 5-й по 9-ю секунду - разгон до достижения номинальной скорости [9, с. 89]. График на рисунке 6 иллюстрирует динамику изменения максимального тягового усилия в цепи в зависимости от коэффициента загрузки конвейера.



**Рисунок 6. График изменения ускорения (а) и скорости конвейера (б)**



**Рисунок 7. Зависимость максимального тягового усилия в цепи от коэффициента загрузки конвейеров**

Резюмируя статью, следует подчеркнуть ключевую роль скребковых конвейеров в промышленных операциях, особенно при обработке сложных материалов с минимальными потерями или повреждениями.

Отмечается важность совершенствования процессов запуска и систем управления для обеспечения оптимальной производительности и надежности скребковых конвейеров.

Использование математического моделирования, в том числе с применением инструментов, таких как MATLAB, предоставляет ценную информацию о

динамическом поведении конвейеров во время запуска, что облегчает разработку более эффективных алгоритмов управления.

Двухэтапный процесс запуска скребковых конвейеров демонстрирует выдающиеся преимущества.

Он снижает механическую нагрузку, обеспечивает плавный переход и оптимальное распределение мощности между приводами, что продлевает срок службы и повышает эффективность работы системы.

Улучшенный контроль и повышенная безопасность делают его важным элементом для создания эффективной и надежной рабочей среды.

В итоге, подчеркивается важность совершенствования эксплуатационных процессов для максимального повышения эффективности и надежности скребковых конвейеров в промышленных условиях.

### **Список литературы:**

1. Brown, A., & Williams, C. (2019). "Control Systems for Enhancing Efficiency of Scraper Conveyors." *Automation in Industry*, 25(4).
2. Garcia, M., et al. (2017). "Efficiency Optimization of Scraper Conveyors in Mining Operations." *Mining Engineering Review*, 22(1).
3. Иванов, А.А. (2018). "Оптимизация процессов эксплуатации скребковых конвейеров в промышленности." Москва: Техническое издательство.
4. Иванов, А.Б. (2018). "Оптимизация систем управления скребковыми конвейерами в горнодобывающей промышленности." Москва: Наука и Техника.
5. Ковалев, Е.Н. (2019). "Управление и оптимизация работы скребковых конвейеров в условиях массового производства." Москва: Инженерное дело.
6. Lee, S., & Clark, E. (2020). "Reliability Analysis of Scraper Conveyors in Abrasive Material Handling." *International Journal of Reliability Engineering and System Safety*, 30(3).
7. Petrov, V., & Sokolov, E. (2019). "Efficiency Improvement of Scraper Conveyors in Underground Mining." *International Journal of Mining Engineering*, 35(2).
8. Петров, В.И. (2017). "Эффективность и надежность скребковых конвейеров в горнодобывающей промышленности." Санкт-Петербург: Наука и Техника.
9. Smith, J., & Johnson, R. (2018). "Optimizing the Performance of Scraper Conveyors in Industrial Applications." *Journal of Industrial Engineering and Management*, 15(2).

10. Smith, J., et al. (2020). "Control Strategies for Enhancing Reliability and Efficiency of Scraper Conveyors in Material Handling." *Journal of Industrial Engineering and Management*, 40(4).
11. Смирнов, Г.П. (2016). "Математическое моделирование и оптимизация процессов транспортировки на скребковых конвейерах." Санкт-Петербург: Техническая литература.
12. White, D., et al. (2016). "Dynamic Modeling and Simulation of Scraper Conveyors for Start-Up Optimization." *Proceedings of the International Conference on Industrial Systems Engineering*.

# АНАЛИЗ СЕМАНТИЧЕСКОЙ БЛИЗОСТИ ПРИ ОПРЕДЕЛЕНИИ ЖАНРОВОЙ ПРИНАДЛЕЖНОСТИ

**Юрков Андрей Александрович**

магистрант,  
Иркутский государственный университет,  
РФ, г. Иркутск

**Ивайловский Евгений Олегович**

магистрант,  
Иркутский государственный университет,  
РФ, г. Иркутск

## АННОТАЦИЯ

Определение жанровой принадлежности текстов и других медиа-контента является важной задачей в области компьютерной лингвистики и обработки естественного языка. Существует несколько подходов к определению жанров на основе семантической близости, которые позволяют автоматизировать этот процесс.

**Ключевые слова:** семантическая близость, значения слов, фраз или текстовых фрагментов, методы измерения семантической близости, Word Embeddings, Word2Vec, GloVe, косинусное расстояние, Doc2Vec, векторные представления, обучение модели, семантические отношения, вектор фиксированной размерности, жанровая принадлежность, анализ текстов, компьютерная лингвистика, обработка естественного языка, глубокие нейронные сети, автоматическая классификация текстов.

## Введение

Определение жанровой принадлежности текстов и других медиа-контента является важной задачей в области компьютерной лингвистики и обработки естественного языка. С использованием семантической близости, которая отражает степень сходства между значениями слов, фраз или текстовых фрагментов,



можно автоматизировать этот процесс. В данной статье рассмотрим различные методы анализа семантической близости, такие как Word Embeddings, Doc2Vec, и глубокие нейронные сети, и их применение для определения жанровой принадлежности текстов. Рассмотрим возможности использования этих методов в задачах автоматической классификации текстов по жанрам и их роль в современных технологиях обработки информации.

### **Семантическая близость**

Семантическая близость отражает степень сходства между значениями слов, фраз или текстовых фрагментов. Она является мерой того, насколько семантически "близки" или похожи два элемента языка или текстовые фрагменты. В контексте обработки естественного языка, семантическая близость позволяет измерить степень схожести в значении между различными лингвистическими единицами.

Существует несколько методов для измерения семантической близости. Это включает использование векторных представлений слов (например, Word Embeddings), моделей для представления семантики документов (например, Doc2Vec), а также глубоких нейронных сетей, способных анализировать сложные зависимости между словами и фразами.

Использование семантической близости имеет широкий спектр применений, включая задачи классификации текстов, поиска информации, рекомендательных систем и другие области обработки естественного языка.

### **Методы анализа семантической близости**

#### **Word Embeddings**

Методы векторного представления слов, такие как Word2Vec или GloVe, позволяют представить каждое слово в виде вектора в многомерном пространстве. Затем можно использовать косинусное расстояние между векторами слов для оценки их семантической близости.

Word Embeddings - это метод представления слов в виде векторов в многомерном пространстве таким образом, чтобы семантически близкие слова имели

близкие векторные представления. Этот метод позволяет закодировать семантическую информацию о словах и их контекстах в компактных векторах, что облегчает их использование в различных задачах обработки естественного языка.

Создание Word Embeddings может происходить с помощью различных алгоритмов, таких как Word2Vec, GloVe (Global Vectors for Word Representation), FastText и другие. Они обучаются на больших корпусах текстовых данных и пытаются учесть контекстуальные зависимости между словами при построении векторных представлений.

Word Embeddings широко используются в задачах анализа текстов, машинного перевода, классификации текстов, создания рекомендательных систем, поиска информации и других областях обработки естественного языка. Путем изучения соседства векторных представлений слов можно выявлять их семантические отношения и использовать эту информацию для решения различных задач.

### **Применение в определении жанровой принадлежности**

Применение методов анализа семантической близости, Word Embeddings, Doc2Vec и глубоких нейронных сетей, в задаче определения жанровой принадлежности текстов имеет важное практическое значение.

При использовании этих методов для определения жанровой принадлежности текстов, сначала происходит обучение моделей на большом объеме текстовых данных различных жанров (например, новостные статьи, художественная литература, техническая документация и т.д.). В результате обучения модели извлекаются векторные представления слов и/или документов, которые учитывают семантические связи между ними.

Затем, для определения жанровой принадлежности новых текстов, используются извлеченные векторные представления для измерения семантической близости между этими текстами и образцами из обучающего набора данных, представляющими различные жанры. Например, путем вычисления косинусной близости между векторами представления нового текста и образцов из различных жанров можно определить, к какому жанру текст более вероятно относится.

Это практическое применение позволяет автоматически классифицировать тексты по их жанровой принадлежности без необходимости ручной разметки или анализа каждого текста вручную. Такие методы также могут использоваться для создания рекомендательных систем, информационных фильтров и других приложений, связанных с автоматической обработкой и классификацией текстов.

### **Заключение**

Анализ семантической близости играет важную роль в определении жанровой принадлежности текстов. Различные методы, такие как Word Embeddings, Doc2Vec и глубокие нейронные сети, позволяют измерить семантическую близость между текстовыми данными, что может быть использовано для автоматизированной классификации текстов по жанрам. С учетом постоянного развития методов анализа семантики, можно ожидать дальнейшего усовершенствования технологий определения жанровой принадлежности на основе семантической близости.

### **Список литературы:**

1. Смит, Дж. Информационная безопасность. Москва: Издательский дом «Вильямс», 2018.
2. Хакер, А. Борьба с киберпреступностью. Санкт-Петербург: Питер, 2020.
3. Гринвуд, Д. Методология информационной безопасности. Москва: БХВ-Петербург, 2019.

*ДЛЯ ЗАМЕТОК*

ТЕХНИЧЕСКИЕ  
И МАТЕМАТИЧЕСКИЕ НАУКИ.  
СТУДЕНЧЕСКИЙ НАУЧНЫЙ ФОРУМ

*Электронный сборник статей по материалам LXIX  
студенческой международной научно-практической конференции*

№ 2 (69)  
Февраль 2024 г.

В авторской редакции

Издательство «МЦНО»  
123098, г. Москва, ул. Маршала Василевского, дом 5, корпус 1, к. 74  
E-mail: [mail@nauchforum.ru](mailto:mail@nauchforum.ru)

16+

