

УДК 004.853

DOI: [10.26102/2310-6018/2026.54.3.014](https://doi.org/10.26102/2310-6018/2026.54.3.014)

## Интеллектуальный поиск патентов-аналогов на основе графовых представлений компонентной структуры изобретения

Д.М. Коробкин✉, С.А. Фоменков, А.Н. Мальков, С.А. Козина

*Волгоградский государственный технический университет, Волгоград,  
Российская Федерация*

**Резюме.** Актуальность работы обусловлена тем, что традиционные системы патентного поиска, основанные на реляционных базах данных и ключевых словах, не способны эффективно учитывать богатый контекст и сложные семантические взаимосвязи, присущие патентным данным. Предложена методика интеллектуального поиска патентов-аналогов на основе изоморфизма подграфов в графовой базе данных, хранящей компонентные структуры устройств, описанных в изобретениях. Интеллектуальность проявляется в способности системы «понимать» структурную суть изобретения, абстрагироваться от текстового описания и находить технически близкие решения даже при несовпадении ключевых слов. Компонентная структура устройств получена посредством анализа патентных текстов с помощью ранее разработанной нейросетевой модели. Используется представление патента в виде графа, где узлы представляют элементы изобретений, а ребра – их взаимосвязи, что позволяет применять графовые алгоритмы для выявления релевантных патентов. Разработаны алгоритмы: парсинга json файла описания компонентной структуры с занесением информации в графовую БД; сравнения графовых представлений компонентной структуры изобретения; редактирования графового представления компонентной структуры изобретения. Практическая значимость заключается в разработанном модуле поиска патентов-аналогов на основе графовых представлений компонентной структуры изобретения, который может быть полезен на этапах формирования патентной заявки патентозаявителем и проведения патентной экспертизы экспертом патентного ведомства. Программный модуль реализован на языке Python с использованием графовой СУБД Neo4j.

**Ключевые слова:** патенты, графовые СУБД, компонентная структура изобретения, сравнение графов, Neo4j.

**Благодарности:** Исследование выполнено при поддержке «Центра цифровых научно-образовательных проектов и разработок в сфере промышленного искусственного интеллекта» Ц2RED-ИИ ВолГТУ, созданного в рамках реализации образовательных программ топ-уровня в сфере искусственного интеллекта (Соглашение № 70-2025-000756).

**Для цитирования:** Коробкин Д.М., Фоменков С.А., Мальков А.Н., Козина С.А. Интеллектуальный поиск патентов-аналогов на основе графовых представлений компонентной структуры изобретения. *Моделирование, оптимизация и информационные технологии*. 2026;14(3). URL: <https://moitvvt.ru/ru/journal/article?id=2117> DOI: 10.26102/2310-6018/2026.54.3.014

## Intellectual search for analogous patents based on graph representations of the invention's structure

D.M. Korobkin✉, S.A. Fomenkov, A.N. Malkov, S.A. Kozina

*Volgograd State Technical University, Volgograd, the Russian Federation*

**Abstract.** The relevance of this work stems from the fact that traditional patent search systems, which are based on relational databases and keywords, are unable to effectively capture the rich context and complex semantic relationships inherent in patent data. The method of intellectual search for patent-

analogous based on subgraph isomorphism in a graph database storing component structures of devices described in inventions is proposed. Intelligence manifests itself in the ability of the system to "understand" the structural essence of the invention, to abstract from the text description and to find technically close solutions even in the case of non-matching keywords. The component structure of the devices was obtained by analyzing patent texts using a previously developed neural network model. A patent is represented as a graph, where nodes represent the elements of the invention and edges represent their relationships, enabling the use of graph algorithms to identify relevant patents. Algorithms have been developed for: parsing a JSON file describing the component structure and loading the information into a graph database; comparing graph representations of the invention's component structure; and editing the graph representation of the invention's component structure. The practical significance lies in the developed patent similarity search module, which is based on graph representations of an invention's component structure. This module can be useful during the stages of filing a patent application by the applicant and conducting a patent examination by a patent office examiner. The software module is implemented in Python using the Neo4j graph DBMS.

**Keywords:** patents, graph DBMS, invention component structure, graph comparison, Neo4j.

**Acknowledgements:** The study was carried out with the support of the Center for Digital Scientific and Educational Projects and Developments in the Field of Industrial Artificial Intelligence (C2RED-AI) of Volgograd State Technical University, created as part of the implementation of top-level educational programs in the field of artificial intelligence (Agreement No. 70-2025-000756).

**For citation:** Korobkin D.M., Fomenkov S.A., Malkov A.N., Kozina S.A. Intellectual search for analogous patents based on graph representations of the invention's structure. *Modeling, Optimization and Information Technology*. 2026;14(3). (In Russ.). URL: <https://moitvvt.ru/ru/journal/article?id=2117> DOI: 10.26102/2310-6018/2026.54.3.014

## Введение

Традиционные системы патентного поиска [1, 2], основанные на реляционных базах данных и ключевых словах, сталкиваются с фундаментальными проблемами синонимии и полисемии [3, 4], не способны эффективно учитывать богатый контекст и сложные семантические взаимосвязи, присущие патентным данным [5].

Необходимо вручную проанализировать текст патентной заявки, выявить ключевые слова (фразы), затем осуществить поиск на основе запросов, сформированных на основе ключевых слов. Например, поисковая система ФИПС Роспатента предполагает поиск на основе запроса к текстовой части документа (реферату, описанию, формуле, названию). При этом используется логический поиск, который позволяет находить документы, содержащий термины запроса, связанные между собой отношениями, определяемыми операторами запроса: AND, OR, NOT, WITHIN, ADJ, \*, (), " ", [ ].

Напротив, графовые СУБД (Neo4j, Amazon Neptune и др.) предлагают новую парадигму<sup>1</sup>, позволяя реализовать семантический и связный поиск, где сам запрос и процесс ранжирования результатов используют силу связей между патентами, изобретателями и технологическими понятиями [6]. Патентные данные обладают врожденной графовой природой, где узлы представляют сущности (патенты, изобретатели, классификации), а ребра – отношения между ними (цитирования, авторство, принадлежность) [7]. В работе [8] используется графовое расширение исходного запроса для повышения полноты поиска.

Ключевое преимущество графовых СУБД – возможность учитывать не только содержание, но и структурную значимость патента. Патент, цитируемый многими другими важными патентами, получает высокий ранг, что указывает на его

<sup>1</sup> Официальная документация Neo4j. URL: <https://neo4j.com/> (дата обращения: 08.09.2025).

фундаментальное влияние в области<sup>2</sup> [9]. Используется понятие «центральности» для выявления наиболее цитируемых патентов и нахождения патентов, являющихся «мостами» между технологическими кластерами [10].

Современные графовые СУБД (например, Neo4j с полнотекстовым индексом от Apache Lucene) поддерживают гибридный поиск: выполняется первоначальный полнотекстовый поиск, затем полученный набор патентов фильтруется и ранжируется с помощью графовых запросов.

В настоящее время наметился передовой край исследований, где графы обогащаются векторными представлениями (эмбедингами). Включение векторных представлений текста патентов, полученных с помощью моделей типа BERT [11] или Sentence-BERT [12, 13], в качестве свойств узлов позволяет осуществлять поиск по семантическому сходству (k-NN поиск). Гибридный семантико-графовый запрос позволяет найти семантически близкие патенты, а затем отфильтровать их по графовым признакам (например, по связи с определенным исследовательским институтом) [14].

Было произведено сравнение графовых СУБД (Neo4j, Amazon Neptune, AllegroGraph, ArangoDB, Memgraph, Dgraph), приведенное в Таблице 1.

Таблица 1 – Результаты сравнения существующих решений  
Table 1 – Results of comparing existing solutions

Критерий\ Система	Neo4j	Amazon Neptune	Allegro-Graph	ArangoDB	Memgraph	Dgraph
Масштабируемость и производительность	Высокая (кластер)	Высокая (управляемый сервис)	Высокая	Средняя	Высокая (в памяти)	Высокая
Стоимость	Коммерческая/ Бесплатная	Зависит от использования AWS	Коммерческая	Open Source (корпоративная версия платная)	Коммерческая/ Бесплатная	Открытый Исходный код
Хранение данных	График свойств	График свойств / RDF	RDF	Multi-Model (Граф, Документ, Key-Value)	График свойств	RDF (GraphQL)
Высокая производительность и поддержка	Да	Да	Да	Да	Да	Да
Графовые алгоритмы	Да (библиотеки)	Да (Gremlin)	Да (SPARQL)	Да (AQL)	Да (Cypher)	Да (GraphQL)
Визуализация данных	Да (Neo4j Bloom, другие инструменты)	Да (с интеграциями)	Да	Да (ArangoDB Oasis)	Да (Memgraph Lab)	Да (Ratel)

В итоге можно сказать, что графовые СУБД открывают новую эру в информационном поиске патентной информации, преодолевая ограничения

<sup>2</sup> Page L., Brin S., Motwani R., Winograd T. *The PageRank Citation Ranking: Bringing Order to the Web*. Wayback Machine. URL: <https://web.archive.org/web/20230329031814/http://www.eecs.harvard.edu/~michaelm/CS222/pagerank.pdf> (дата обращения: 08.09.2025).

традиционных систем. Синтез графовых алгоритмов, семантических векторных представлений и методов машинного обучения позволяет создавать интеллектуальные системы, которые не только находят релевантные патенты, но и выявляют скрытые закономерности, прогнозируют тренды и проактивно рекомендуют информацию. Несмотря на существующие вызовы, именно гибридные графово-семантические подходы являются наиболее перспективным направлением для создания следующего поколения систем интеллектуальной аналитики.

### Материалы и методы

Интеллектуальный поиск патентов-аналогов на основе графовых представлений компонентной структуры изобретения — это подход, при котором изобретение моделируется в виде графа, где вершины соответствуют компонентам (элементам, узлам, деталям и т. д.), а рёбра — связям между ними (механическим, пространственным, атрибутивным и т. д.). Поиск аналогов сводится к сравнению таких графов или их фрагментов.

Интеллектуальность проявляется не просто в автоматизации, а в способности системы «понимать» структурную суть изобретения, абстрагироваться от текстового описания и находить технически близкие решения при несовпадении ключевых слов.

В работе [15] описан нейросетевой подход к извлечению компонентной структуры изобретения из патентных документов, содержащей сущности (изобретение, его компоненты и их характеристики) и пять типов взаимосвязей (Рисунок 1).

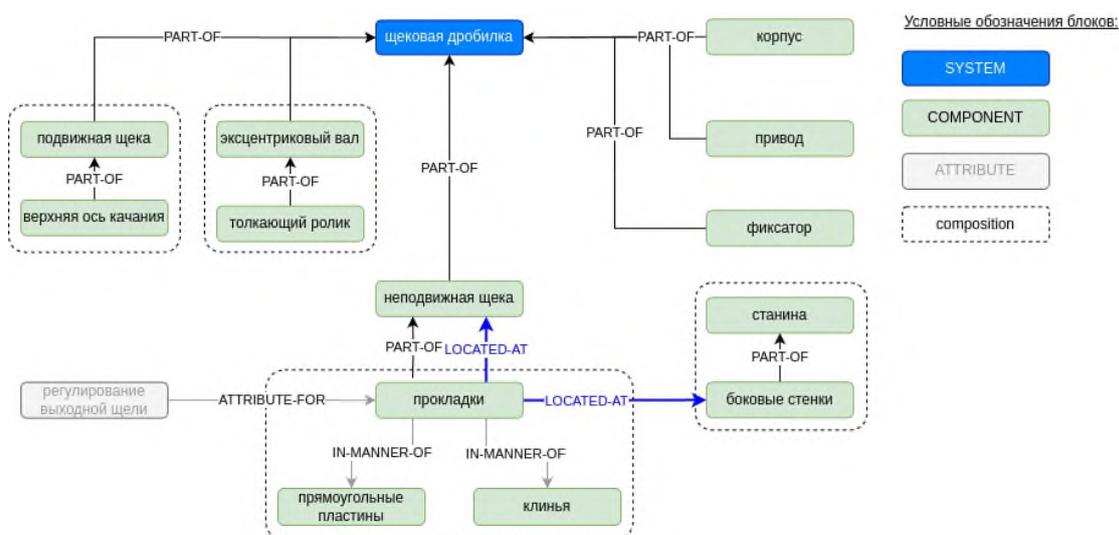


Рисунок 1 – Компонентная структура изобретения  
Figure 1 – Component structure of the invention

Такая графовая структура хранится в виде JSON файла (Рисунок 2), получаемого как результат анализа патентного текста, содержащего описание компонентной структуры изобретения, с помощью обученной нейросетевой модели.

```
"document": "Система для рециркуляции выхлопных газов, которая включает в себя впускной коллектор, канал рециркуляции выхлопных газов",
"tokens": [{"text": "впускной коллектор", "token_start": 10, "token_end": 11, "entityLabel": "COMPONENT"},
{"text": "Система для рециркуляции выхлопных газов", "token_start": 0, "token_end": 4, "entityLabel": "SYSTEM"},
{"text": "канал рециркуляции выхлопных газов", "token_start": 13, "token_end": 16, "entityLabel": "COMPONENT"},
{"text": "впускным коллектором", "token_start": 20, "token_end": 21, "entityLabel": "COMPONENT"},
{"text": "клапан рециркуляции выхлопных газов", "token_start": 25, "token_end": 28, "entityLabel": "COMPONENT"},
{"text": "корпус датчика", "token_start": 32, "token_end": 33, "entityLabel": "COMPONENT"},
{"text": "впускным коллектором", "token_start": 38, "token_end": 39, "entityLabel": "COMPONENT"},
{"text": "чувствительный элемент", "token_start": 42, "token_end": 43, "entityLabel": "COMPONENT"},
{"text": "чувствительный элемент", "token_start": 49, "token_end": 50, "entityLabel": "COMPONENT"},
{"text": "измерения абсолютного давления", "token_start": 45, "token_end": 47, "entityLabel": "ATTRIBUTE"},
{"text": "измерения дифференциального давления", "token_start": 52, "token_end": 54, "entityLabel": "ATTRIBUTE"},
{"text": "впускным коллектором", "token_start": 61, "token_end": 62, "entityLabel": "COMPONENT"}],
"relations": [{"child": 0, "head": 10, "relationLabel": "PART-OF"},
{"child": 0, "head": 13, "relationLabel": "PART-OF"},
{"child": 20, "head": 13, "relationLabel": "CONNECTED-WITH"},
{"child": 20, "head": 25, "relationLabel": "PART-OF"},
{"child": 0, "head": 32, "relationLabel": "PART-OF"},
{"child": 38, "head": 32, "relationLabel": "CONNECTED-WITH"},
{"child": 38, "head": 42, "relationLabel": "PART-OF"},
{"child": 38, "head": 49, "relationLabel": "PART-OF"},
{"child": 42, "head": 45, "relationLabel": "ATTRIBUTE-FOR"}]
```

Рисунок 2 – Структура JSON файла  
 Figure 2 – JSON file structure

Предлагаемый автоматизированный процесс основан на преобразовании структурированных данных в граф, где узлы представляют элементы изобретений, а ребра – их взаимосвязи, что позволяет применять графовые СУБД для хранения патентной информации, а их поисковые возможности – для выявления релевантных патентов.

Цель работы заключается в разработке программы поиска патентов-аналогов на основе сравнений графовых представлений компонентной структуры изобретения.

*Разработка алгоритмов.* Алгоритм парсинга json-файла (Рисунок 3) описания компонентной структуры необходим для автоматизации процесса добавления в графовую базу данных сущностей и связей.

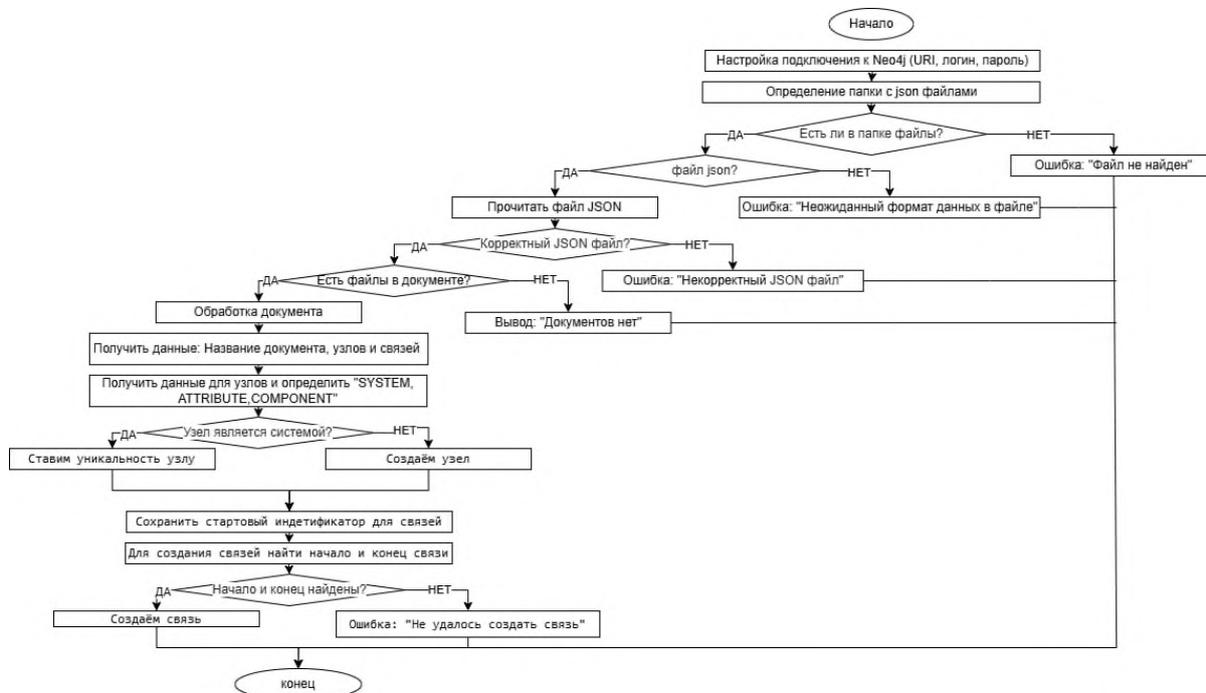


Рисунок 3 – Алгоритм парсинга json файла  
 Figure 3 – JSON file parsing algorithm

В цикле обработки для каждого токена извлекаются его текст, начальная позиция и метка сущности: SYSTEM, COMPONENT, ATTRIBUTE, создается узел графа. Полученный идентификатор узла сохраняется в словаре token\_nodes.

После завершения цикла обработки токенов начинается цикл по связям. Для каждой связи извлекаются начальные позиции дочернего и головного токенов, а также тип связи: PART-OF, LOCATED-AT, CONNECTED-WITH, IN-MANNER-OF, ATTRIBUTE-FOR. Эти позиции сопоставляются с идентификаторами узлов из словаря token\_nodes. Если обе позиции найдены, добавляется связь между соответствующими узлами.

Алгоритм сравнения графовых представлений (Рисунок 4) компонентной структуры изобретения описывает процесс сравнения двух документов (патентов) между собой, генерации отчета о совпадениях и визуализации общих элементов при помощи Neo4j.

После инициализации происходит извлечение всех токенов и сохранение полученных данных во внутреннюю коллекцию documents.

Пользователь может устанавливать имена двух документов для сравнения. Выполняется функция compare\_documents (doc1, doc2), в которой осуществляется сравнение по child (дочерний узел), head (родительский узел) и label (тип связи). Результаты анализа возвращаются в виде словаря matches. На основе него формируется отчет о совпадениях, который выводится пользователю.

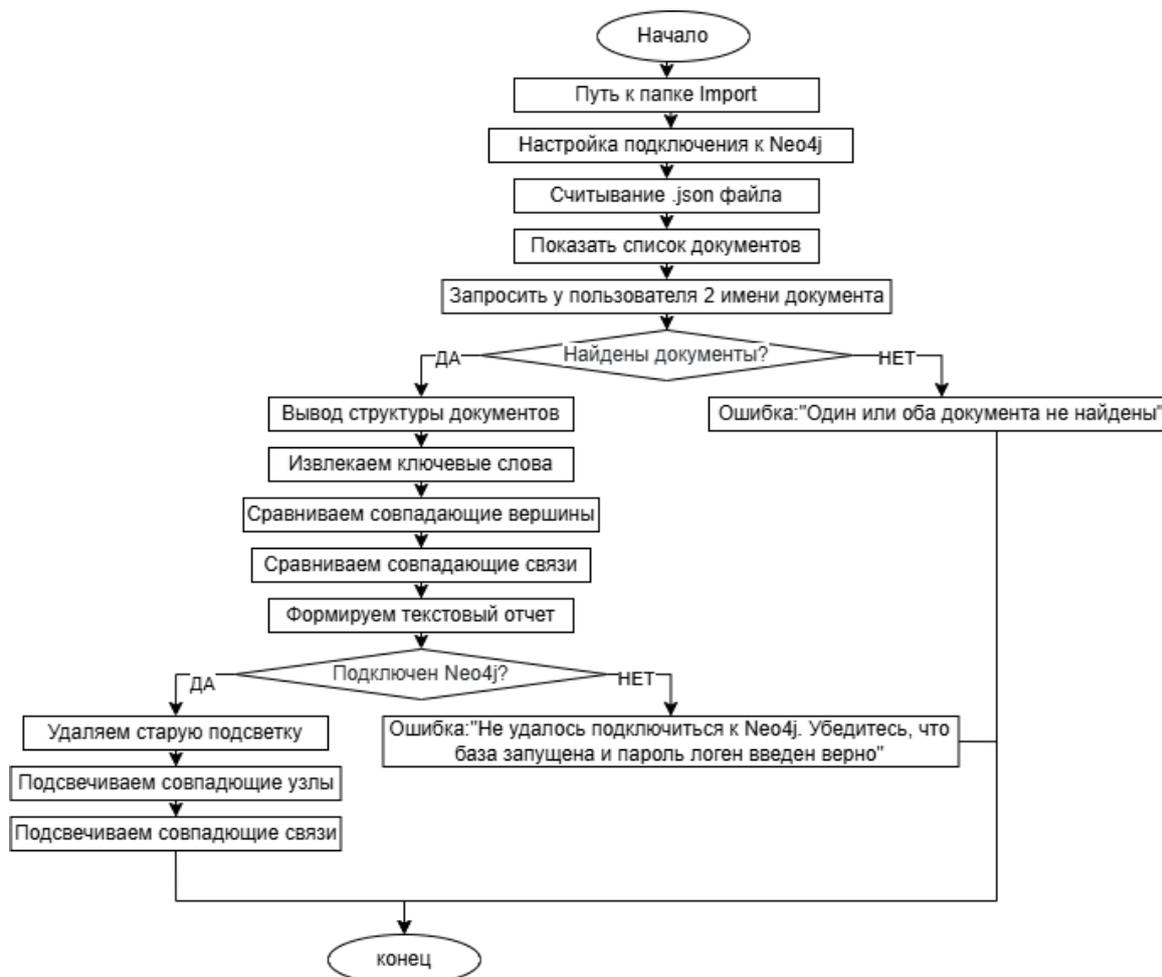


Рисунок 4 – Алгоритм сравнения графовых представлений  
Figure 4 – Graph representation comparison algorithm

Разработан математический аппарат поиска изоморфизма подграфов (Subgraph Isomorphism) в графовой базе данных, хранящей компонентные структуры устройств, описанных в изобретениях.

Целевой граф:

$$G_T = (V_T, E_T, L_V, L_E),$$

где  $V_T$  – множество вершин (компоненты устройства);  $E_T \subseteq V_T \times V$  – множество ребер (связи между компонентами);  $L_V: V_T \rightarrow \sum V$  – функция меток вершин, которая ставит в соответствие каждой вершине ее тип (например, «двигатель», «вал», «пружина») из алфавита  $\sum V$ ;  $L_E: E_T \rightarrow \sum E$  – функция меток ребер, которая ставит в соответствие каждому ребру его тип («CONNECTED\_WITH», «PART-OF», «LOCATED\_AT») из алфавита  $\sum E$ .

Паттерн-граф (образец):  $G_P = (V_P, E_P, L_V, L_E)$ , где множества и функции определены аналогично.

Определение: Паттерн-граф  $G_P$  изоморфен некоторому подграфу целевого графа  $G_T$ , если существует инъективная функция (отображение)  $f: V_P \rightarrow V_T$ , обладающая следующими свойствами:

а) Сохранение меток вершин:

$$\forall u \in V_P: L_V(u) = L_V(f(u)).$$

Это условие гарантирует, что компонент определенного типа (например, «электродвигатель») отображается только на компонент того же типа в целевом патенте.

б) Сохранение структуры и связей:

Для каждого ребра в паттерн-графе должно существовать соответствующее ребро между образами вершин в целевом графе:

$$\forall (u, v) \in E_P: (f(u), f(v)) \in E_T.$$

Должны также совпадать типы связей (метки ребер):

$$\forall (u, v) \in E_P: L_E(u, v) = L_E(f(u), f(v)).$$

в) Условие инъективности:

Разные вершины паттерна отображаются в разные вершины целевого графа:

$$\forall u, w \in V_P, u \neq w: f(u) \neq f(w).$$

Данное условие исключает случай, когда два разных компонента паттерна «склеиваются» в один компонент в целевом патенте, что обычно недопустимо для сравнения устройств.

Алгоритм редактирования графового представления (Рисунок 5) компонентной структуры изобретения использует визуальный интерфейс Neo4j Bloom. Графовые данные, созданные на основе компонентных структур изобретений, можно найти по названию изобретения, его уникальному идентификатору или по ключевым словам, содержащимся в узлах графа.

Редактирование узлов включает в себя несколько типовых действий. Пользователь может изменить текстовое название узла, метку узла, создать новые узлы вручную, указав их текстовое значение и тип (метку), чтобы дополнить или уточнить структуру изобретения.

Связи между узлами также поддаются редактированию. Изменение типа существующих связей доступно вручную и через свойства relationship, доступные в интерфейсе Neo4j Bloom.

Проверка целостности графа позволяет обеспечить логическую связность и корректность модели перед дальнейшим анализом или экспортом данных.

*Проектирование программного модуля.* Основными компонентами программы являются: модуль парсера JSON-файла, модуль загрузки данных в графовую БД, база данных Neo4j, модуль сравнения графов, модуль визуализации и редактирования графов.

Модуль парсера JSON файла реализует следующие функции: парсинг – позволяет пользователю загрузить файл в формате JSON, содержащий описание компонентной структуры; проверка корректности структуры загруженного JSON-файла, включая соответствие формату и наличие всех обязательных элементов. Модуль загрузки данных в графовую БД осуществляет автоматическую загрузку сущностей и связей в Neo4j. Модуль сравнения графов реализует сравнение совпадающих вершин и связей. Модуль визуализации обеспечивает подсветку совпадающих вершин и связей в Neo4j, а также формирование итогового отчета о сравнении, включающего все найденные совпадения. Модуль редактирования графов позволяет просматривать и редактировать графы через интерфейс визуализации, а изменения при необходимости сохраняются в базу данных.

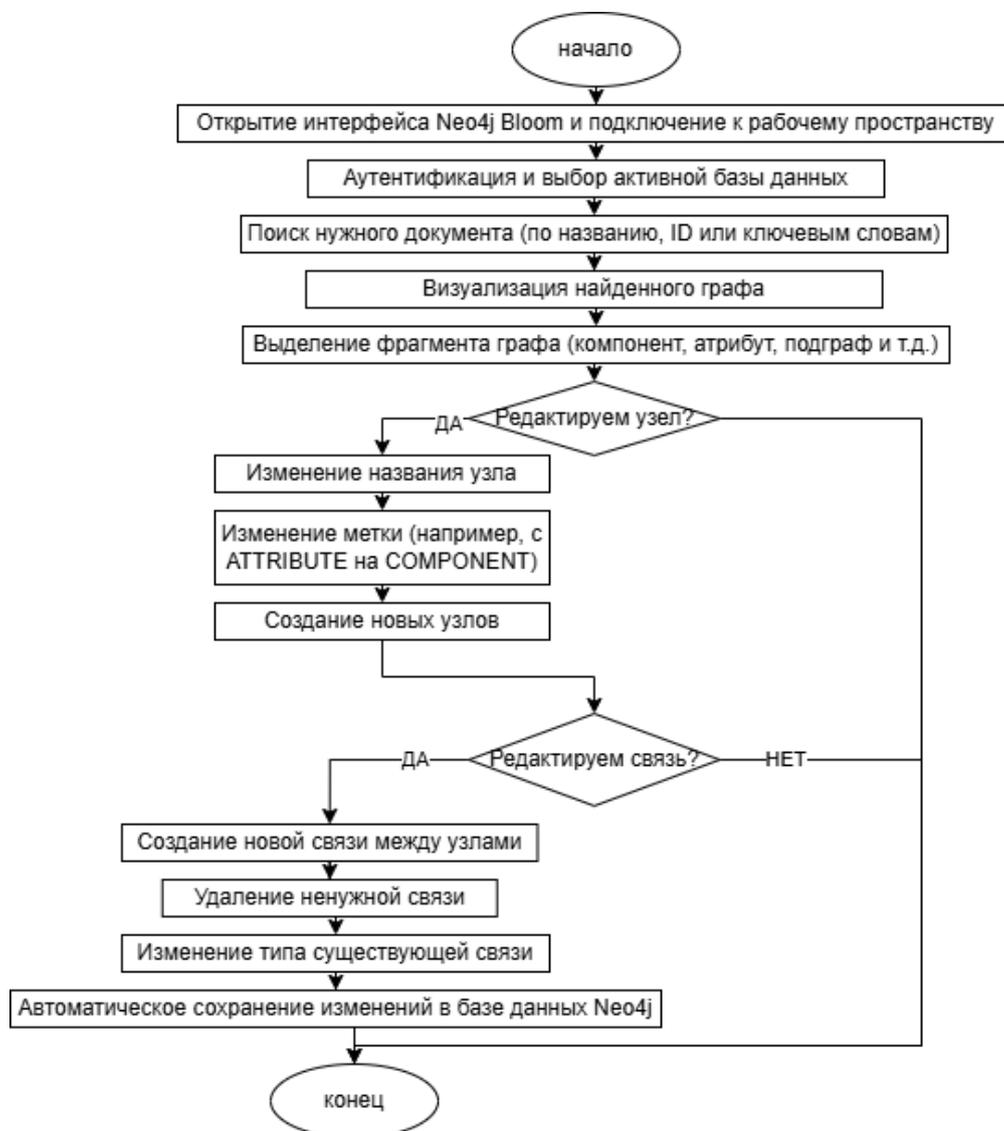


Рисунок 5 – Алгоритм редактирования графа  
Figure 5 – Graph editing algorithm

*Программная реализация.* Программный модуль реализован с использованием языка программирования Python, который обеспечивает как обработку входных данных, так и взаимодействие с графовой базой данных Neo4j. Визуализация графов (Рисунок 6) осуществляется средствами самой среды Neo4j, куда данные автоматически заносятся программой и где пользователь может визуально анализировать совпадения, выделенные цветом. Используется библиотека SpaCy для морфологического анализа текстовых данных (приведения словоформ к лемме).

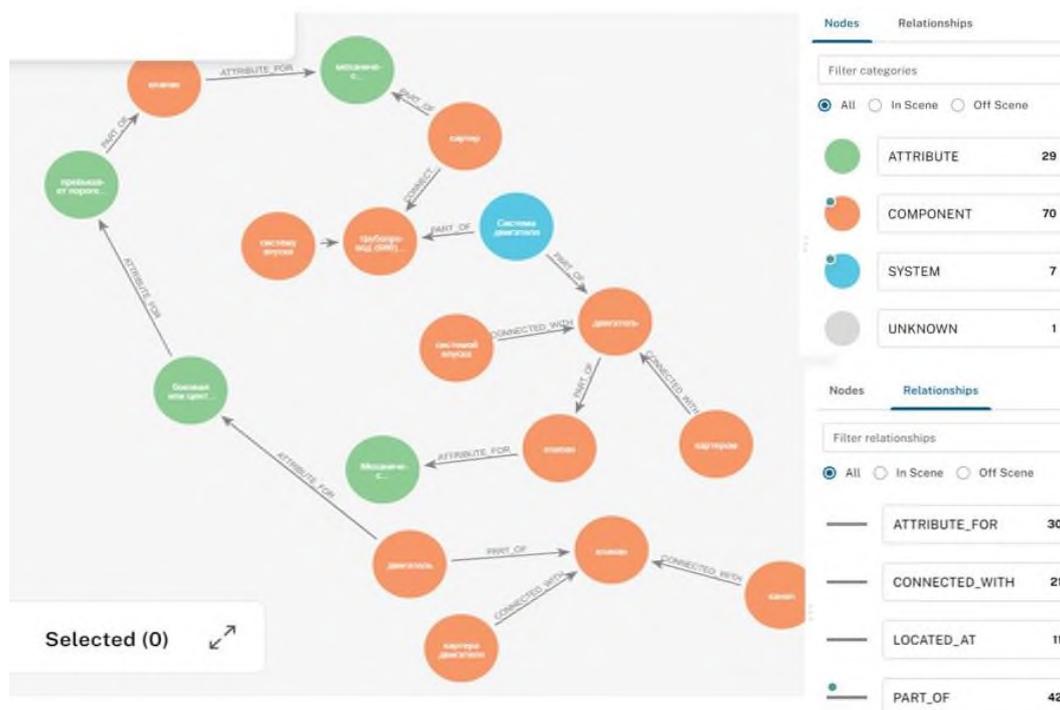


Рисунок 6 – Графовое представление в Neo4j  
Figure 6 – Graph representation in Neo4j

При сравнении патентов (Рисунок 7) программный модуль эффективно выявляет общие вершины и совпадающие отношения. Результаты совпадали с ожидаемыми эталонными выборками, что подтверждает правильность алгоритма сравнения и его пригодность для патентного поиска.

```
Общие токены (вершины): ['центральные тела', 'Насос', 'эластичных тел', 'Электромагниты', 'Электромагниты']
Общие связи:
{'child': 'насос', 'head': 'центральные тела', 'relationLabel': 'part-of'}
{'child': 'эластичных тел', 'head': 'центральные тела', 'relationLabel': 'located-at'}
```

Рисунок 7 – Сравнение графовых представлений 2-х патентов  
Figure 7 – Comparison of graph representations of two patents

Подсветка совпадающих узлов и отношений в интерфейсе Neo4j Browser представлена на Рисунке 8. При совпадении вершин и связей программа автоматически выделяет их красным цветом.

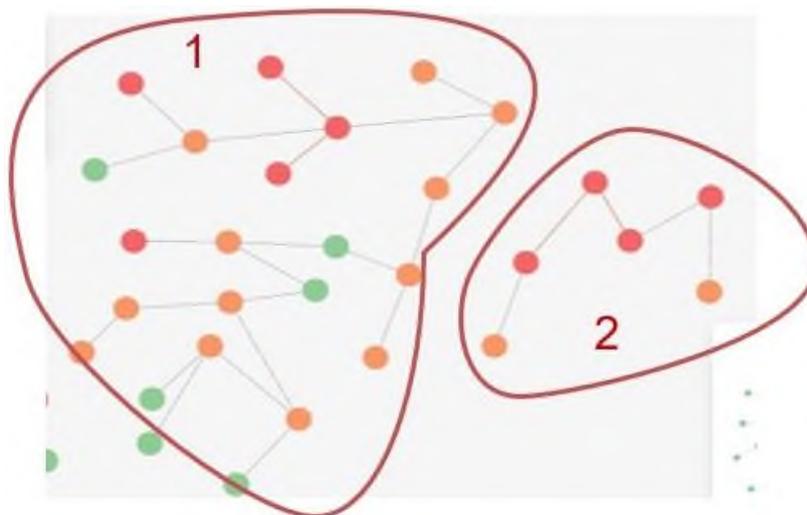


Рисунок 8 – Визуализация совпадений  
Figure 8 – Visualization of matches

### Результаты

Достигнуты следующие результаты:

- проанализированы описания компонентной структуры запатентованного устройства (системы), изучены графовые БД;
- разработан алгоритм парсинга json файла описания компонентной структуры с занесением информации в графовую БД;
- сформирован алгоритм сравнения графовых представлений компонентной структуры изобретения;
- разработан алгоритм редактирования графового представления компонентной структуры изобретения;
- спроектирован, программно реализован модуль и протестирована его работоспособность на тестовых примерах.

### Заключение

Предложена методика поиска патентов-аналогов на основе изоморфизма подграфов (Subgraph Isomorphism) в графовой базе данных, хранящей компонентные структуры устройств, описанных в изобретениях. Используется представление патента в виде графа, где узлы представляют элементы изобретений, а ребра – их взаимосвязи, что позволяет применять графовые алгоритмы для выявления релевантных патентов.

Практическая значимость заключается в разработанном модуле поиска патентов-аналогов на основе графовых представлений компонентной структуры изобретения, который может быть полезен на этапах формирования патентной заявки патентозаявителем и проведения патентной экспертизы экспертом патентного ведомства.

### СПИСОК ИСТОЧНИКОВ / REFERENCES

1. Фоменков С.А., Коробкин Д.М., Коробкина В.С. Поиск патентов-аналогов на основе сравнения ключевых понятий. *Инженерный вестник Дона*. 2024;(11). URL: <http://www.ivdon.ru/ru/magazine/archive/n11y2024/9604>

- Fomenkov S.A., Korobkin D.M., Korobkina V.S. Search for patent analogues based on a comparison of key phrases. *Engineering Journal of Don*. 2024;(11). (In Russ.). URL: <http://www.ivdon.ru/en/magazine/archive/n11y2024/9604>
2. Бобунов А.В., Коробкин Д.М., Фоменков С.А., Васильев С.С. Разработка программного модуля поиска патентов-аналогов. *Инженерный вестник Дона*. 2022;(11). URL: <http://www.ivdon.ru/ru/magazine/archive/n11y2022/8018>  
Bobunov A.V., Korobkin D.M., Fomenkov S.A., Vasiliev S.S. Development of a software module for searching for patent analogues. *Engineering Journal of Don*. 2022;(11). (In Russ.). URL: <http://www.ivdon.ru/en/magazine/archive/n11y2022/8018>
  3. Baeza-Yates R., Ribeiro-Neto B. *Modern Information Retrieval: The Concepts and Technology Behind Search*. Addison-Wesley; 2011. 913 p.
  4. Liu B. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*. Berlin, Heidelberg: Springer; 2011. 624 p. <https://doi.org/10.1007/978-3-642-19460-3>
  5. Kim W.Ch., Mauborgne R. *Blue Ocean Strategy: How to Create Uncontested Market Space and Make Competition Irrelevant*. Boston: Harvard Business School Press; 2005. 240 p.
  6. Robinson I., Webber J., Eifrem E. *Graph Databases: New Opportunities for Connected Data*. Sebastopol: O'Reilly Media; 2015. 238 p.
  7. Hogan A., Blomqvist E., Cochez M., et al. Knowledge Graphs. *ACM Computing Surveys*. 2021;54(4). <https://doi.org/10.1145/3447772>
  8. Sarica S., Song B., Low E., Luo J. Engineering Knowledge Graph for Keyword Discovery in Patent Search. *Proceedings of the Design Society: International Conference on Engineering Design*. 2019;1(1):2249–2258. <https://doi.org/10.1017/dsi.2019.231>
  9. Haveliwala T.H. Topic-Sensitive PageRank. In: *WWW '02: Proceedings of the 11<sup>th</sup> International Conference on World Wide Web, 07–11 May 2002, Honolulu, HI, USA*. New York: ACM; 2002. P. 517–526. <https://doi.org/10.1145/511446.511513>
  10. McColl R.C., Ediger D., Poovey J., Campbell D., Bader D.A. A performance evaluation of open source graph databases. In: *PPAA '14: Proceedings of the first workshop on Parallel programming for analytics applications, 16 February 2014, Orlando, FL, USA*. New York: ACM; 2014. P. 11–18. <https://doi.org/10.1145/2567634.2567638>
  11. Devlin J., Chang M.-W., Lee K., Toutanova K. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. arXiv. URL: <https://arxiv.org/abs/1810.04805> [Accessed 9<sup>th</sup> September 2025].
  12. Reimers N., Gurevych I. *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks*. arXiv. URL: <https://arxiv.org/abs/1908.10084> [Accessed 9<sup>th</sup> September 2025].
  13. Yang Y., Cui X. Bert-Enhanced Text Graph Neural Network for Classification. *Entropy*. 2021;23(11). <https://doi.org/10.3390/e23111536>
  14. Zuo H., Yin Y., Childs P. Patent-KG: Patent Knowledge Graph Extraction for Engineering Design. *Proceedings of the Design Society*. 2022;2:821–830. <https://doi.org/10.1017/pds.2022.84>
  15. Korobkin D., Vasiliev S., Fomenkov S., Kravets A. The Deep Learning Method for Information Extracting about the Components of Patented Devices and their Relationships. In: *2024 6<sup>th</sup> International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency (SUMMA), 13–15 November 2024, Lipetsk, Russia*. IEEE; 2024. P. 437–442. <http://doi.org/10.1109/SUMMA64428.2024.10803690>

## ИНФОРМАЦИЯ ОБ АВТОРАХ / INFORMATION ABOUT THE AUTHORS

**Коробкин Дмитрий Михайлович**, кандидат технических наук, доцент, Волгоградский государственный технический университет, Волгоград, Российская Федерация.

*e-mail:* [dkorobkin80@mail.ru](mailto:dkorobkin80@mail.ru)

ORCID: [0000-0002-4684-1011](https://orcid.org/0000-0002-4684-1011)

**Dmitry M. Korobkin**, Candidate of Engineering Sciences, Associate Professor, Volgograd State Technical University, Volgograd, the Russian Federation.

**Фоменков Сергей Алексеевич**, доктор технических наук, профессор, Волгоградский государственный технический университет, Волгоград, Российская Федерация.

*e-mail:* [saf550@yandex.ru](mailto:saf550@yandex.ru)

ORCID: [0000-0001-9907-4488](https://orcid.org/0000-0001-9907-4488)

**Sergey A. Fomenkov**, Doctor of Engineering Sciences, Professor, Volgograd State Technical University, Volgograd, the Russian Federation.

**Мальков Андрей Николаевич**, магистрант, Волгоградский государственный технический университет, Волгоград, Российская Федерация.

*e-mail:* [anmalkov11@gmail.com](mailto:anmalkov11@gmail.com)

**Andrey N. Malkov**, Master's Degree student, Volgograd State Technical University, Volgograd, the Russian Federation.

**Козина Светлана Александровна**, инженер-исследователь, Волгоградский государственный технический университет, Волгоград, Российская Федерация.

*e-mail:* [ksvetlan54@gmail.com](mailto:ksvetlan54@gmail.com)

ORCID: [0000-0003-4049-620X](https://orcid.org/0000-0003-4049-620X)

**Svetlana A. Kozina**, Research Engineer, Volgograd State Technical University, Volgograd, the Russian Federation.

*Статья поступила в редакцию 07.11.2025; одобрена после рецензирования 12.03.2026; принята к публикации 25.03.2026.*

*The article was submitted 07.11.2025; approved after reviewing 12.03.2026; accepted for publication 25.03.2026.*