

УДК 004.89

doi: 10.15622/rcai.2025.071

## АВТОМАТИЧЕСКОЕ ПОСТРОЕНИЕ ТЕЗАУРУСА: ОЦЕНКА МЕТОДОВ КЛАСТЕРИЗАЦИИ И ВЕКТОРНОГО ПРЕДСТАВЛЕНИЯ СЛОВ

И.А. Коломойцева (*bolatiger@mail.ru*)

О.И. Федяев (*olegfedyayev@mail.ru*)

Донецкий национальный технический университет, Донецк

В статье исследуются методы автоматизированного построения тезауруса для построения системы интеллектуального поиска информации «CodeLex», с акцентом на выявление синонимии между терминами в области программирования. Рассматривается применение алгоритмов кластеризации (K-means, DBSCAN) в сочетании с моделями векторного представления слов (Word2Vec, BERT) для определения отношения синонимии. Эксперименты проводились на фрагментах текстов из книги по языку программирования PHP. Результаты показали, что связка Word2Vec+K-means наиболее перспективна для определения синонимии между именами функций и их описаниями. BERT продемонстрировал низкую эффективность в данной задаче. Полученные результаты могут быть использованы для улучшения работы системы CodeLex.

**Ключевые слова:** информационный поиск, тезаурус, кластеризация, DBSCAN, K-means, Word2Vec, BERT.

### Введение

Поиск учебных и научных материалов – актуальная задача для любого учебного заведения. Первое, что используется с этой целью как источник информации, – это библиотека, в которой в современных условиях хранятся не только бумажные источники, но и электронные. Ещё одним таким источником материалов может быть структурное подразделение учебного заведения, например, кафедра вуза. В кафедральных хранилищах могут быть представлены книги, монографии, учебные пособия, методические указания и другие материалы, созданные сотрудниками кафедры. Также в кафедральных хранилищах есть большое количество служебных документов. Искать информацию в этих источниках могут преподаватели.

даватели, которым нужно подготовить лекцию, обновить некоторые факты и т.п. К этим же ресурсам могут обращаться и студенты, которым нужно найти ответы на вопросы, возникающие в процессе обучения. Поиск в таких локальных хранилищах с помощью разработанной в рамках проекта «Виртуальная кафедра» системы интеллектуального поиска информации в области программирования «CodeLex» является важной задачей.

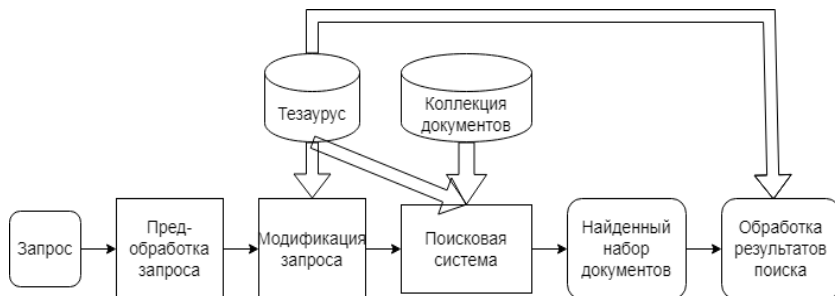


Рис. 1. Схема системы интеллектуального поиска информации «CodeLex»

В системе «CodeLex» интеллектуального поиска в документах образовательной организации (рис. 1), кроме типовых элементов, есть принципиально важный дополнительный – тезаурус, ориентированный на предметную область «Программирование».

Тезаурус – это словарь, в котором слова и словосочетания с близкими значениями сгруппированы в единицы, называемые понятиями, и в котором явно (в виде отношений, иерархии) указываются отношения между этими понятиями [Лукащевич, 2011].

Тезаурус участвует в трёх аспектах функционирования системы «CodeLex». Во-первых, с его помощью модифицируется (расширяется или уточняется) запрос пользователя. Это позволяет увеличить полноту, а в случае уточнения запроса, и точность поиска. Тезаурус также участвует в построении инвертированного индекса – важной составляющей «движка» поисковой системы. Элементами такого индекса выступают термины из тезауруса, а не полный набор терминов из коллекции документов, с которой работает поисковая система. Подобный подход ускорит поиск и сократит место в памяти для хранения индекса. Кроме того, тезаурус используется в алгоритме ранжирования документов. Благодаря тезаурусу, более высокий ранг имеют документы, включающие термины, связанные отношениями с терминами из запроса.

Сложность работы с тезаурусом в предметной области программирования заключается в том, что его надо сформировать, а затем постоянно поддерживать в актуальном состоянии.

Существуют два основных подхода к созданию тезаурусов: ручной и автоматизированный. Ручной подход – создание тезауруса экспертами-лингвистами или специалистами предметной области. Он трудоёмкий и дорогостоящий. Автоматизированный подход – использование методов обработки естественного языка (NLP), машинного обучения и алгоритмов кластеризации. Этот метод позволяет быстро обрабатывать большие объёмы данных, но нуждается в последующей ручной проверке и коррекции.

Целью данной статьи является исследование методов автоматизированного построения тезаурусов, в частности, использование алгоритмов кластеризации для определения отношения синонимии между терминами.

## **1. Анализ подходов к автоматизированному построению тезаурусов**

Автоматический подход к созданию тезаурусов использует методы машинного обучения и обработки естественного языка (NLP) для автоматизации различных этапов создания. И так как размеченных текстов в области программирования недостаточно, то предпочтительными являются методу обучения без учителя.

Для автоматизированного построения тезауруса чаще всего используются методы:

- кластеризации;
- снижения размерности;
- векторного представления слов;
- тематического моделирования;
- ассоциативных правил.

Рассмотрим особенности применения этих методов для задачи формирования тезаурусов.

Кластеризация выполняет группировку терминов по семантической близости. Например, кластеры могут соответствовать категориям тезауруса (напр., «ООП», «Сетевое программирование<sup>2</sup>). Кластеры также позволяют выявить группы связанных терминов. Иерархическая кластеризация дополнительно строит древовидную структуру, что соответствует иерархиям в тезаурусе (родо-видовые отношения).

Среди методов кластеризации для решения поставленной задачи подходящими методами представляются следующие:

- K-means;
- иерархическая кластеризация;
- DBSCAN (Density-Based Spatial Clustering of Applications with Noise).

K-means разбивает данные на K кластеров, где каждый элемент относится к кластеру с ближайшим средним (центроидом).

Иерархическая кластеризация строит иерархию кластеров, начиная с каждого элемента как отдельного кластера и постепенно объединяя их, или наоборот, начиная со всего набора данных как одного кластера и постепенно разделяя его.

DBSCAN находит кластеры, основанные на плотности точек данных, выделяя "шум" – точки, не принадлежащие ни к одному кластеру.

Перед кластеризацией или анализом, чтобы упростить данные или улучшить их визуализацию, применяют следующие методы снижения размерности: PCA, t-SNE, UMAP.

PCA может быть использован для предварительной обработки данных, чтобы снизить размерность и улучшить производительность алгоритмов кластеризации. Его основным недостатком является то, что он не гарантирует сохранения локальной структуры данных.

t-SNE подходит для визуализации тезауруса в 2D или 3D пространстве, позволяя оценить качество кластеризации. Его недостаток – может искажать глобальную структуру данных.

UMAP обеспечивает лучшее сохранение как локальной, так и глобальной структуры данных, что делает его хорошим выбором для снижения размерности перед кластеризацией.

Векторные представления слов используются для вычисления семантической близости терминов (обнаружения синонимов, ассоциации). При обучении без учителя используются модели Word2Vec, FastText, BERT.

Тематическое моделирование – это статистический метод, позволяющий извлечь скрытые темы из большой коллекции текстовых документов. Он основан на предположении, что каждый документ является смесью нескольких тем, и каждая тема состоит из распределения слов. Вместо ручного анализа текста, тематическое моделирование автоматически идентифицирует эти темы и определяет, как каждая тема представлена в каждом документе. Самыми популярными методами тематического моделирования являются методы: латентное размещение Дирихле (LDA), и латентный семантический анализ (LSA). LSA позволяет выявить в текстах синонимию и полисемию, и поэтому его важно исследовать в контексте построения тезауруса.

Ассоциативные правила могут выявлять отношения между словами на основе их совместной встречаемости в текстах. Эти правила могут обнаружить такие типы отношений, как родо-видовые, часть-целое, а также связанные понятия.

Методы, использующие ассоциативные правила, которые можно использовать в контексте построения тезауруса: Apriori, Eclat.

## 2. Методы кластеризации и векторного представления слов

В данной статье исследуется эффективность применения моделей построения векторного представления слов – Word2Vec и BERT и методов кластеризации – K-means и DBSCAN [Жилов, 2023].

Word2Vec для формирования векторного представления слов использует нейронные сети с малым числом слоёв. На вход нейросети подается большой текстовый корпус, в котором каждому слову сопоставляется вектор. После создания словаря вычисляется векторное представление слов, основанное на семантической близости. Мерой близости является косинусное сходство между двумя не нулевыми векторами [Попова, 2023].

Bidirectional Encoder Representations from Transformers (BERT) – нейронная сеть, основанная на энкодере трансформера. BERT используется для решения задач обработки естественно-языковых текстов, в том числе, и для построения векторного представления слов. Архитектура модели BERT организована в виде цепочки блоков. Первому блоку на вход подаётся цепочка эмбеддингов, а на выходе получается последовательность векторов той же длины [Новикова, 2024].

Метод K-means [Дюличева, 2021] является алгоритмом кластеризации, который разбивает данные на k кластеров, минимизируя расстояние внутри кластера. Он в теории не предназначен для поиска синонимов, но его можно использовать для определения близких по смыслу слов. Чтобы его использовать для поиска синонимов, необходимо создать векторное представление слов с помощью методов Word2Vec или FastText или получить контекстные эмбеддинги с помощью моделей трансформеров, например, BERT. Далее необходимо применить алгоритм K-means к полученным векторным представлениям слов. Каждый кластер, сформированный K-means, должен представлять собой группу слов, семантически близких друг к другу.

DBSCAN – это метод кластеризации, основанный на плотности данных [Фомичев, 2023]. При использовании не надо предварительно указывать количество кластеров. Метод DBSCAN может обнаруживать кластеры произвольной формы. Как и метод K-means, DBSCAN по умолчанию не предназначен для поиска синонимов, но при определенных условиях его можно использовать для этой цели. Как и для K-means, для DBSCAN потребуются векторные представления слов (или контекстные эмбеддинги), на основе которых DBSCAN и выдаст кластеры близких по смыслу слов. В отличие от K-means, для которого требуется задать количество кластеров, DBSCAN нужно задать радиус окрестности (eps), в котором будут искаться точки для формирования кластеров, и минимальное количество точек (min\_samples), необходимое для формирования "плотного" кластера, чтобы снизить размерность и улучшить производительность алгоритмов.

### 3. Оценка эффективности Word2Vec и BERT в задачах кластеризации синонимов идентификаторов и их описаний

Структура тезауруса с области программирования, используемого в CodeLex, приведена в [Коломойцева, 2024]. Одной из важных особенностей этого тезауруса является то, что в качестве синонимов могут выступать имена функций (методов класса), классов, структур. Возможность автоматически расширить запрос за счёт добавления имени идентификатора увеличит точность поиска.

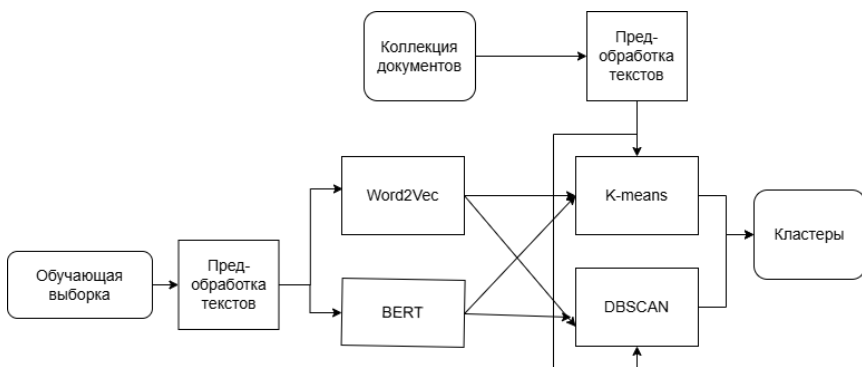


Рис. 2. Схема экспериментальной оценки эффективности задачи кластеризации синонимов

Синонимическую связь между названием идентификатора и термином, определяющим его назначение можно установить с помощью алгоритмов кластеризации. В качестве таких алгоритмов для исследования выбраны K-means и DBSCAN. Так как эти алгоритмы работают с числовым представлением информации, то потребовалось векторное представление слов. Для реализации этой задачи решено использовать модели Word2Vec и BERT.

Задачей исследования является определение эффективности использования связок моделей и методов Word2Vec+K-means, Word2Vec+DBSCAN, BERT+K-means, BERT+DBSCAN (рис. 2).

Эффективность измеряется по формуле

—

где  $K_H$  – количество имён идентификаторов и их описаний, попавших в один кластер, т.е. являющихся синонимами,  $K_C$  – количество имён идентификаторов и их описаний, содержащихся в коллекции документов.

Эксперименты проводились с помощью программы на языке Python. Для предобработки текста использовалась библиотека NLTK, для лемматизации терминов – rymorphy2. Реализация методов K-means и DBSCAN взята из библиотеки scikit-learn. Работа с моделью Word2Vec выполняется с помощью gensim. С BERT работа выполняется с помощью transformers, и в качестве наиболее подходящей модели выбрана мультязычная модель bert-base-multilingual-cased, позволяющая работать с текстами на русском языке.

Выборкой, на которой обучались модели Word2Vec и BERT, были фрагменты текстов из книги по программированию на языке PHP, содержащие описание работы со строками и массивами.

Перед обучением моделей тексты прошли процесс предобработки, который включает следующие этапы:

- разбиение текста на токены;
- удаление неинформативных токенов;
- приведение слов на русском языке к нижнему регистру;
- лемматизация токенов.

К неинформативным токенам относятся стоп-слова (из набора стоп-слов для русского языка из библиотеки NLTK), числа (в том числе и числа с дополнительными символами, например, #7), аргументы функций, имена переменных. Также удалены слова общей лексики, которые не могут быть терминами из области программирования, например, «также», «существовать», «избегать». Такие «лишние» слова определены экспертным путём и содержатся в дополнительном словаре.

К нижнему регистру приводятся слова для обеспечения унификации. Применяется это только к словам на русском языке из-за того, что слова, написанные латиницей, могут быть именами идентификаторов, для которых регистр имеет смысл.

Эффективность методов проверялась на двух наборах текстов, каждый из которых содержал примерно по 3800 слов. Набор1 содержал описание 19 функций работы со строками, а набор2 – описание 12 функций работы с массивами.

Эксперимент выявил, что использование модели BERT для векторного представления слов, привело к тому, что все слова, записанные латиницей, оказались в одном кластере. И ни одно русскоязычное описание не попало в этот кластер. При этом оказалось неважно, каким методом выполнялась кластеризация – K-means или DBSCAN.

При использовании для векторного представления слов модели Word2Vec DBSCAN разбил тексты на кластеры таким образом, что для набора1 и набора 2 нашлось по 3 верных сопоставления. В тоже время алгоритм K-means для набора1 составил кластеры таким образом, что

совпали имена функций и их описания в 13 случаях, а для набора2 – в 10. Расчёт эффективности для различных комбинаций моделей векторного представления слов и алгоритмов кластеризации представлен в табл. 1.

Во время эксперимента использовались такие параметры модели Word2Vec:

- размерность вектора слов – 100 (варьировались от 100 до 300);
- размер окна контекста – 4 (варьировались от 3 до 7);
- минимальная частота слова – 1;
- количество потоков – 4;
- алгоритм обучения CBOW (Continuous Bag of Words), т.е. модель пытается восстановить центральное слово по окружающим словам.

Остальные параметры использовали значение по умолчанию, предоставляемые моделью Word2Vec из библиотеки `gensim`.

Размерность выходных векторов для модели `bert-base-multilingual-cased` 768. Модель BERT переводится в режим оценки с помощью `model.eval()`. Это необходимо, чтобы отключить слои, которые используются только во время обучения. При токенизации слов с использованием BERT tokenizer добавлены параметры `truncation=True` и `padding=True`. Это необходимо для обработки случаев, когда входные слова длиннее максимальной длины последовательности, поддерживаемой BERT, и для обеспечения одинаковой длины всех входных последовательностей.

Число кластеров для K-means установлено опытным путём равным 15. Все остальные параметры использовали значение по умолчанию, установленные для этого метода библиотекой `scikit-learn`.

Для модели DBSCAN:

- максимальное расстояние между двумя точками, при котором они считаются соседними друг другу, `eps=0.065`;
- минимальное количество соседей, необходимое точке, чтобы стать центром кластера, `min_samples=4`.

Таблица 1

	Word2Vec+ K-means	Word2Vec+ DBSCAN	BERT+ K-means	BERT+ DBSCAN
Набор1	0,68	0,16	0	0
Набор2	0,83	0,25	0	0

Данный эксперимент показал, что для определения синонимии между именем функции и её описанием наиболее перспективной оказалась связка Word2Vec+K-means, обеспечивающая наилучший баланс между точностью определения синонимических связей и вычислительной эффективностью, что делает её подходящим кандидатом для использования в системе CodeLex. Однако, полученные результаты не являются окончательными и требуют дальнейшей проверки на большем объеме данных и с применением различных стратегий предобработки текста.



## Заключение

Проведенное исследование подтвердило перспективность автоматического построения тезауруса для предметной области программирования, в частности, для автоматического выявления синонимических связей между идентификаторами их текстовыми описаниями. Экспериментально установлено, что комбинация методов Word2Vec и K-means показала наилучшие результаты в сравнении с другими исследованными подходами (Word2Vec+DBSCAN, BERT+K-means, BERT+DBSCAN). Однако, несмотря на относительно высокую эффективность связки Word2Vec+K-means (достигающей 0.68 и 0.83 для разных наборов данных), полученные результаты нельзя считать окончательными, и они требуют дальнейшего улучшения и углубленного анализа.

Низкая эффективность BERT в данной задаче требует особого внимания. Вероятной причиной неудовлетворительной работы модели является отсутствие адаптации (fine-tuning) на специализированном корпусе данных, включающем примеры кода и документации по программированию. Предварительно обученные модели BERT, как правило, оптимизированы для работы с общими текстами и могут испытывать трудности с обработкой специфического вокабуляра и синтаксиса языков программирования. Также возможно, что мультязычная модель BERT недостаточно хорошо адаптирована к сочетанию русского языка (в описаниях) и латиницы (в именах идентификаторов), что привело к обособлению латинских терминов в отдельные кластеры.

Для повышения качества автоматического построения тезауруса следует предпринять ряд шагов.

Во-первых, сравнить использование Word2Vec и BERT с моделью векторного представления слов FastText. FastText, в отличие от Word2Vec, использует информацию о морфологии слов. Это может быть полезно для обработки словоформ в описаниях функций и для более адекватной обработки имён идентификаторов.

Во-вторых, выполнить fine-tuning BERT на специализированных данных, т.е. адаптировать модель BERT к предметной области программирования путем fine-tuning на корпусе, состоящем из исходного кода, документации, комментариев и других текстовых материалов, связанных с языком PHP и другими популярными языками программирования. После fine-tuning необходимо повторно провести эксперименты с BERT+K-means и BERT+DBSCAN, чтобы оценить влияние адаптации модели на качество кластеризации.

Также актуальной задачей является расширение обучающей выборки и анализ ошибок. Вместе с тем важно оптимизировать параметры кластеризации. В дальнейшем, кроме эмпирического подбора параметров, следует применить методы автоматической оптимизации, например, метод локтя (Elbow method) для определения количества кластеров для K-means.

Для расширения тезауруса необходимо исследовать возможность использования тематического моделирования (например, LDA или NMF) для выявления скрытых тем и семантических связей между терминами в области программирования и применения методов, основанных на ассоциативных правилах,

Таким образом, проведенное исследование можно использовать в качестве основы для построения тезауруса в области программирования системы интеллектуального поиска информации «CodeLex».

### Список литературы

- [Дюличева, 2021] Дюличева Ю.Ю. Учебная аналитика MOOK как инструмент анализа математической тревожности // Вопросы образования. – 2021. – № 4. – С. 243-265. – doi 10.17323/1814-9545-2021-4-243-265.
- [Жилов, 2023] Жилов Р.А. Интеллектуальные методы кластеризации данных // вестия Кабардино-Балкарского научного центра РАН. – 2023. – № 6(116). – С. 152-159. – DOI 10.35330/1991-6639-2023-6-116-152-159.
- [Коломойцева, 2024] Коломойцева И.А. Особенности структуры тезауруса для повышения качества поиска документов в области программирования // Программная инженерия: методы и технологии разработки информационно-вычислительных систем (ПИИВС-2024): Сборник материалов и докладов V Международной научно-практической конференции, Донецк, 27–28 ноября 2024 года. – Донецк: Донецкий национальный технический университет, 2024. – С. 245-250.
- [Лукашевич, 2011] Лукашевич Н.В. Тезаурусы в задачах информационного поиска. – М.: Изд-во Московского университета, 2011. – 512 с.
- [Новикова, 2024] Новикова О.А., Ермолов А.Е. Сравнительный анализ эффективности работы алгоритмов кластеризации текстов // Перспективы науки. – 2024. – № 5(176). – С. 24-31.
- [Попова, 2023] Попова О.А. Анализ методов векторизации текстовых документов // Вестник Рязанского государственного радиотехнического университета. – 2023. – № 85. – С. 96-102. – doi: 10.21667/1995-4565-2023-85-96-102.
- [Фомичев, 2023] Фомичев, Д. А. Кластеризация вакансий по их описанию с использованием машинного обучения и методов анализа текста // Международная конференция по мягким вычислениям и измерениям. – 2023. – Т. 1. – С. 201-204.