МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»
Кафедра математической кибернетики и компьютерных наук

МЕТОД ТЕМАТИЧЕСКОГО ПОИСКА НА ОСНОВЕ ТЕОРИИ ГРУБЫХ МНОЖЕСТВ

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 5 курса 551 группы
направления 09.03.04 — Программная инженерия
факультета КНиИТ
Джумакулова Романа Маратовича

Научный руководитель
доцент, к. ф.-м. н. _____ С. В. Папшев

Заведующий кафедрой
к. ф.-м. н. _____ С. В. Миронов

Саратов 2026

ВВЕДЕНИЕ

В последние годы объём текстовой информации, с которой ежедневно взаимодействует человек, неуклонно возрастает. Новостные статьи, научные публикации, посты в социальных сетях порождают терабайты неструктурированных данных. Эффективная обработка и поиск среди этих массивов становятся критически важной задачей. Традиционные методы поиска, основанные на ключевых словах, часто оказываются недостаточно гибкими из-за явлений синонимии и полисемии: одни и те же слова в разном контексте могут иметь разные смыслы, а разные слова могут означать одно и то же. В связи с этим особую значимость приобретают методы, способные учитывать семантическую близость документов и расширять поисковые запросы.

Настоящая работа посвящена созданию прототипа тематической поисковой системы для русскоязычного корпуса новостных документов с применением теории грубых множеств. В основе работы лежат методы векторного представления текста, алгоритмы кластеризации и подход к расширению поисковой выдачи на основе теории грубых множеств. В качестве языка реализации выбран Python с использованием библиотек scikit-learn, gensim, NLTK и pymorphy. Все компоненты поисковой системы позволяют выполнять предобработку текстов, векторизацию, кластеризацию, формирование поисковой выдачи с выделением rough-set областей и оценку качества по метрикам Precision@K, NDCG@K и MRR.

Объектом исследования выступает процесс тематического поиска текстовых документов, предметом — методы векторизации, кластеризации и расширения поискового запроса на основе теории грубых множеств.

Цель выпускной квалификационной работы — создать прототип поисковой системы с применением теории грубых множеств. Для достижения указанной цели в ходе работы решаются следующие задачи:

- Изучить способы векторного представления, кластеризации текстовых данных, а также теорию грубых множеств и метрики оценки результатов информационного поиска.
- Провести серию экспериментов по выбору наилучшего метода векторизации по итогам кластеризации и информационного поиска.
- С методом - лидером провести дополнительный эксперимент, приближенный к реальному поиску.

— Проанализировать полученные результаты, сделать вывод.

В ходе работы использованы методы контент-анализа научной и практической литературы, объектно-ориентированное программирование, инструментальные эксперименты в среде Python, а также сравнительный анализ полученных результатов.

Практическая значимость работы проявляется в создании воспроизводимого прототипа тематической поисковой системы, который может быть адаптирован для поиска по корпусам русскоязычных текстов в различных предметных областях, а также использован в учебном процессе для демонстрации методов обработки естественного языка и информационного поиска.

Структура и объём работы. Работа состоит из введения, двух разделов, заключения, списка использованных источников и четырёх приложений. Общий объём работы — 52 страницы, из них 35 страниц — основное содержание, список использованных источников информации — 27 наименований.

1 Методы векторизации, кластеризации и теория грубых множеств

Современные задачи информационного поиска и обработки естественного языка требуют преобразования текстовой информации в формат, пригодный для математических операций. Основным способом такого преобразования является векторное представление, при котором каждый документ описывается вектором признаков. Это позволяет применять методы машинного обучения, определять смысловую близость между текстами через расстояние между векторами и выполнять их тематическую группировку.

1.1 Методы векторизации

В работе рассматриваются четыре основных подхода к построению векторных представлений текстов:

- **Bag-of-words (мешок слов)** — один из наиболее ранних методов, представляющий текст как набор слов без учёта их взаимного расположения. Достоинством метода является простота реализации, недостатками — высокая размерность и разреженность пространства, а также игнорирование семантических связей между словами.
- **TF-IDF (term frequency–inverse document frequency)** — статистический показатель, оценивающий важность слова в документе относительно всего корпуса. Чем чаще слово встречается в документе и чем реже во всём корпусе, тем больший вес ему присваивается. Данный метод позволяет частично решить проблему стоп-слов, однако остаётся уязвимым к явлению синонимии.
- **Латентно-семантический анализ (LSA)** — метод, который сочетает TF-IDF-взвешивание с последующим снижением размерности пространства с помощью сингулярного разложения (SVD). Это позволяет выделить скрытые семантические связи между терминами и документами, уменьшить влияние шумовых признаков и получить более компактное векторное пространство.
- **Doc2Vec** — нейросетевой метод, кодирующий документ в плотное векторное пространство малой размерности (обычно от 100 до 500 измерений). Существует в двух архитектурах: Distributed Memory (PV-DM), учитывающей локальный контекст слов, и Distributed Bag of Words (PV-DBOW), предсказывающей случайные слова из документа только по его вектору.

Для всех методов векторизации требуется единая предобработка текстовых данных, включающая нормализацию (приведение к нижнему регистру, удаление пунктуации и чисел), токенизацию (разбиение на отдельные слова), удаление стоп-слов (местоимений, союзов, предлогов) и стемминг или лемматизацию (приведение слов к единой форме). В работе используется лемматизация с помощью библиотеки pymorphy2.

1.2 Кластеризация текстовых данных

Кластеризация представляет собой форму обучения без учителя, предназначенную для разделения документов на подмножества (кластеры) таким образом, что документы внутри одного кластера оказываются похожи друг на друга, но максимально отличаются от содержимого других кластеров. В работе рассматриваются два алгоритма:

— **Метод К-средних** — итеративный алгоритм, в котором задаётся начальное количество кластеров k k, случайным образом выбираются центры кластеров, после чего документы распределяются по ближайшим центрам, а центры пересчитываются как средние значения всех документов кластера. Процесс повторяется до минимизации среднеквадратичного отклонения.

— **Метод К-медоид** — концептуально схож с К-средними, но в качестве центра кластера (медоида) выбирается реальный документ, что делает алгоритм более устойчивым к выбросам. Именно этот метод используется в практической части работы для кластеризации векторных представлений.

Выбор количества кластеров осуществляется с помощью метода локтя (elbow method), основанного на анализе зависимости внутрикластерной ошибки от числа кластеров.

Для оценки качества кластеризации используются три внутренние метрики:

— **Коэффициент силуэта (Silhouette Score)** — показывает, насколько объект похож на объекты своего кластера и отделён от ближайшего чужого кластера. Значения лежат в диапазоне от -1 до 1, где чем ближе к 1, тем лучше.

— **Индекс Калински-Харабаж (Calinski-Harabasz Index)** — оценивает отношение межкластерного разброса к внутрикластерному. Чем выше значение, тем более плотными и хорошо разделёнными являются кластеры.

— **Индекс Девиса-Болдина (Davies-Bouldin Index)** — оценивает среднюю схожесть каждого кластера с наиболее близким к нему другим кластером. Чем меньше значение, тем лучше.

1.3 Теория грубых множеств

Теория грубых множеств, предложенная польским математиком Здиславом Павлаком, основывается на утверждении, что в рамках ограниченного набора признаков объекты невозможно различить при совпадении этих признаков. Если некоторое множество объектов не может быть точно описано через классы эквивалентности, оно разбивается на две аппроксимации:

— **Нижняя аппроксимация** — объекты, которые на основании имеющихся характеристик можно точно отнести к множеству X X.

— **Верхняя аппроксимация** — объекты, которые потенциально могут принадлежать множеству X X.

— **Граничная область** — разность между верхней и нижней аппроксимациями, содержащая объекты с неоднозначной принадлежностью.

В контексте данного исследования универсальным множеством являются текстовые документы коллекции, а множество X X — документы, релевантные поисковому запросу. Поскольку релевантность не всегда является однозначной, такое множество рассматривается как приближённое, что позволяет расширять поисковую выдачу за счёт включения документов из граничной области и верхней аппроксимации.

1.4 Метрики оценки поисковой выдачи

Для оценки качества информационного поиска используются следующие метрики:

— **Precision@K** — доля релевантных документов среди первых K K результатов выдачи. В работе используются два варианта: строгий (релевантность = 2) и мягкий (релевантность больше или равна 1).

— **NDCG@K (Normalized Discounted Cumulative Gain)** — метрика, учитывающая не только наличие релевантных документов в выдаче, но и их позицию. Чем выше расположен релевантный документ, тем больше его вклад в итоговый показатель. Значение нормируется на идеальное ранжирование и находится в диапазоне от 0 до 1.

— **MRR (Mean Reciprocal Rank)** — показывает, насколько рано в выдаче появляется первый строго релевантный документ. Рассчитывается как среднее обратных позиций первого релевантного документа по всем запросам.

Оценка релевантности выполняется экспертом по трёхуровневой шкале: 0 — не релевантно, 1 — частично релевантно, 2 — полностью релевантно.

2 Практическая реализация методов векторизации, кластеризации и поисковой системы

Для реализации описанных методов был выбран язык программирования Python. В ходе разработки использовались следующие библиотеки: pandas для работы с табличными данными, NumPy для численных операций, scikit-learn для TF-IDF-векторизации, снижения размерности (TruncatedSVD) и расчёта метрик кластеризации, gensim для Doc2Vec, NLTK и pymorphy2 для обработки русского языка, umap-learn для визуализации, Matplotlib для построения графиков.

2.1 Используемый корпус текстовых документов

В работе используется корпус русскоязычных текстов, состоящий из более чем 29 тысяч новостных статей с сайта Саратовского государственного университета. Корпус размещён в Excel-файле, где первый столбец содержит токенизированные и лемматизированные тексты новостей, прошедшие предварительную обработку, а второй — исходные тексты. Общей темой всех новостей является университет, его факультеты, преподаватели и студенты, что несколько затрудняет задачу кластеризации, но наглядно демонстрирует важность расширения поискового запроса.

2.2 Предобработка текстовых данных

Реализована функция токенизации с удалением стоп-слов:

— текст приводится к нижнему регистру;
— с помощью регулярного выражения выделяются отдельные токены;
— из списка исключаются стоп-слова (союзы, предлоги, местоимения);
— удаляются токены, состоящие только из цифр.

Список стоп-слов включает типичные для русского языка служебные части речи, а также специфические шумовые токены.

2.3 Векторизация текстовых данных

2.3.1 Векторизация с помощью TF-IDF

Реализована функция $\mathrm{vectorize\_tfidf}$, использующая $\mathrm{TfidfVectorizer}$ из scikit-learn с параметрами:

— $\mathrm{ngram\_range}=(1,\ 2)$ — учитываются униграммы и биграммы;
— $\mathrm{min\_df}=2$ — термин должен встретиться минимум в двух документах;
— $\mathrm{max\_df}=0.70$ — исключаются термины, встречающиеся более чем в 70

— max_features=30000 — ограничение размера словаря;
— sublinear_tf=True — логарифмическое масштабирование частоты;
— norm="l2" — евклидова нормализация векторов.

Результатом является разреженная TF-IDF-матрица размера (число документов) * (30 000 признаков).

2.3.2 Векторизация с помощью LSA

Реализована функция vectorize_lsa, которая выполняет следующие шаги:

1. построение TF-IDF-матрицы с описанными выше параметрами;
2. снижение размерности с помощью TruncatedSVD (число компонент выбирается как минимум из заданного значения, числа документов минус один и числа признаков минус один);
3. L2-нормализация полученных векторов с помощью Normalizer.

Итоговая матрица LSA-векторов имеет размерность (число документов) ? (число латентных компонент), где документы, близкие по тематике, оказываются ближе друг к другу в векторном пространстве.

2.3.3 Векторизация с помощью Doc2Vec

Реализована функция vectorize_doc2vec, обучающая модель gensim.models.Doc2Vec с параметрами:

— vector_size=100 — размерность вектора документа;
— window=5 — размер контекстного окна;
— min_count=2 — исключение слишком редких слов;
— epochs=30 — количество проходов по корпусу;
— dm=1 — использование архитектуры Distributed Memory.

После обучения векторы документов извлекаются из модели и нормализуются по L2-норме.

2.4 Кластеризация документов

Для кластеризации используется метод K-медоид, реализованный в библиотеке scikit-learn-extra. Реализованы две функции:

— make_kmedoids_model — создаёт модель с фиксированными параметрами (метрика — косинусное расстояние, метод оптимизации — pam, инициализация — k-medoids++);

— cluster_with_kmedoids — выполняет кластеризацию, возвращает метки кластеров, индексы медоидов и статистику размеров кластеров.

Для оценки качества кластеризации реализована функция compute_internal_clustering_metrics, вычисляющая три метрики:

— коэффициент силуэта (с косинусной метрикой);
— индекс Калински-Харабаж;
— индекс Девиса-Болдина.

Сравнение методов векторизации по этим метрикам показало превосходство метода LSA по всем трём компонентам. LSA сочетает преимущества TF-IDF и снижения размерности, формирует более компактное и менее разреженное пространство признаков и оказывается более устойчивым, чем Doc2Vec, который требует обучения нейросетевой модели на последовательностях токенов.

2.5 Поисковая система на основе теории грубых множеств

Разработан алгоритм тематического поиска, включающий следующие этапы:

1. **Преобразование запроса.** Поисковый запрос проходит ту же предобработку, что и документы корпуса, после чего преобразуется в векторное пространство выбранной модели (TF-IDF, LSA или Doc2Vec).
2. **Расчёт близости.** Вычисляется косинусная близость между вектором запроса и векторами всех документов коллекции.
3. **Выделение rough-set областей.** Для каждого кластера рассчитывается доля документов, близких к запросу (попадающих в верхние 10 процентов по косинусной близости). На основе этого значения кластер относится к одной из трёх областей:
   — **нижняя аппроксимация** — доля больше или равно 0.25;
   — **граничная область** — доля от 0.05 до 0.25;
   — **внешняя область** — доля меньше 0.05.
4. **Формирование выдачи.** Итоговая поисковая выдача формируется как объединение:
   — прямых результатов (документы с наибольшей косинусной близостью к запросу);
   — документов из нижней аппроксимации;
   — документов из граничной области.

При этом исключаются дубликаты документов.

Программный код реализованных функций размещён в приложениях к работе.

2.6 Анализ полученных результатов

Для выбора лучшего метода векторизации было проведено сравнение TF-IDF, LSA и Doc2Vec. Результаты показали, что метод LSA превосходит остальные по коэффициенту силуэта, индексу Калински-Харабаж и индексу Девиса-Болдина. Это объясняется тем, что LSA сохраняет статистическую основу TF-IDF, но дополнительно снижает размерность пространства признаков и выделяет скрытые тематические связи между документами.

Для закрепления результатов была проведена экспертная оценка поисковой выдачи для пяти тематических запросов. Метод LSA также показал наилучшие результаты.

2.7 Итоговый эксперимент

В рамках итогового эксперимента был сформирован набор из 40 поисковых запросов, представляющих собой заголовки новостных статей, случайным образом выбранные из коллекции. По итогам 40 запросов была получена выдача из 1867 документов. После проведения экспертной оценки и подсчёта метрик были получены следующие результаты:

- **Precision@10 (оценки 1-2)** = 0.83 — высокая доля хотя бы частично релевантных документов в первых десяти результатах;
- **NDCG@10** = 0.68 — релевантные документы в целом располагаются достаточно высоко в выдаче;
- **Precision@10 (оценка 2)** = 0.19 — доля строго релевантных документов ниже;
- **MRR** = 0.47 — первый строго релевантный документ находится в среднем на 2-3 позиции.

Полученные значения показывают, что разработанная поисковая система находит тематически близкие документы, но требует дальнейшей настройки для повышения точности. Система эффективна для расширенного тематического поиска, когда пользователю важно получить набор близких по смыслу материалов, а не только точное совпадение с формулировкой запроса.

ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы был разработан прототип тематической поисковой системы для корпуса русскоязычных новостных документов с применением теории грубых множеств.

По результатам сравнения методов векторизации по трём внутренним метрикам кластеризации наилучшие результаты показал метод LSA. Это объясняется тем, что LSA сохраняет статистическую основу TF-IDF, но дополнительно снижает размерность пространства признаков и выделяет скрытые тематические связи между документами.

Итоговый эксперимент на 40 контрольных запросах показал следующие результаты:

- Precision@10 для частично и полностью релевантных документов (оценки 1-2) составил 0.83;
- NDCG@10 составил 0.68;
- Precision@10 для строго релевантных документов (оценка 2) составил 0.19;
- MRR составил 0.47.

Полученные значения показывают, что разработанная поисковая система хорошо находит тематически близкие документы, но требует дальнейшей настройки для повышения точности строгого соответствия запросу. Предложенный подход эффективен для расширенного тематического поиска, когда пользователю важно получить набор близких по смыслу материалов, а не только точное совпадение с формулировкой запроса.

В процессе работы были получены практические навыки обработки естественного языка, векторизации текстовых данных, кластеризации, применения теории грубых множеств в задаче информационного поиска, а также оценки качества поисковых систем.