Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»
КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К КУРСОВОЙ РАБОТЕ

НА ТЕМУ:

«Компилятор языка Oberon»

Студент группы **ИУ7-11М** _____ **Е.В. Брянская**
(Подпись, дата) (И.О.Фамилия)

Руководитель _____ **А.А. Ступников**
(Подпись, дата) (И.О.Фамилия)

2023 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Согласно ежегодному прогнозу, который опубликовала Международная корпорация данных (IDC, International Data Corporation [1]), занимающаяся мониторингом количества информации, объём созданных и воспроизведённых данных в 2020 году резко вырос [2].

Установлено, что этот показатель достиг 64.2 Збайт (10^{21} байт) данных, что примерно в 2 раза больше, чем в 2018 году, когда отметка достигла 33 Збайт [2, 3]. Также предполагается, что в период с 2020 по 2025 годы объём будет продолжать активно расти [4].

В связи с этим задача поиска необходимой информации в больших массивах данных обостряется с каждым годом всё больше. Проблема усугубляется ещё и тем, что один и тот же объект может описываться по-разному, также стоит учитывать, что одна группа людей его может употреблять в ином контексте, нежели другая, всё это создаёт дополнительные сложности в обработке текстов на естественном языке.

Целью данной выпускной квалификационной работы является разработка и реализация метода определения объекта из ограниченной выборки по нечёткому описанию на русском языке.

Для достижения цели необходимо решить следующие задачи:

- изучить основные алгоритмы компьютерной лингвистики и обосновать выбор тех, которые будут использованы для модернизации;
- сформировать выборку респондентов для формирования исходного перечня терминов;
- разработать метод определения объекта из ограниченной выборки по нечёткому описанию на русском языке;
- разработать программное обеспечение и протестировать его;
- оценить работоспособность разработанного метода и дать рекомендации о его применимости.

1 Аналитическая часть

1.1 Формализация задачи

В соответствии с темой необходимо разработать метод, позволяющий по нечёткому описанию на русском языке определить объект, принадлежащий готовой ограниченной выборке.

В рамках поставленной задачи в качестве выборки используются понятия из словаря терминов в области ДВС [5], список выбранных слов указан в приложении А. Для того, чтобы обеспечить работоспособность метода, необходимо заранее собрать как можно больше определений выбранных терминов.

На вход подаётся словесное описание какого-либо термина из этого набора. Оно представляется в виде множества слов, связанных между собой по смыслу и грамматически, на русском языке.

Результатом является термин, заданное описание которого наиболее совпадает с введённым. Термин представляет из себя одно слово или словосочетание.

Для выполнения данной задачи требуется выполнить следующие шаги:

1) предварительно подготовить терминологическую базу знаний, в основе которой лежат сущности конкретной предметной области и их множественные определения;
2) полученное входное описание сравнить с имеющимися в базе знаний;
3) по результатам сравнения выбрать наиболее подходящий термин.

1.2 Возможные области применения

Подобный метод может быть применён в медицинских системах, ориентированных на определение вида заболевания с последующим подбором наиболее целесообразного способа лечения по описанию симптомов, результатов обследований и т.д.

Также он может быть привлечён в системах контроля и оценки знаний

учащихся, в частности, в заданиях с развёрнутым ответом, для оценивания правильности которого привлекаются специалисты. Разрабатываемый подход может позволить освободить часть сотрудников от этой работы, и снизить роль человеческого фактора. О необходимости снижения которого не раз заявляли руководители Федеральной службы по надзору в сфере образования и науки. Снижение этого показателя стало одной из причин создания ЕСОКО (единая система оценки качества образования).

Кроме того, метод может быть полезен при повышении квалификации или переквалификации работников, поскольку в любой сфере деятельности складывается определённая динамическая система понятий, термины которой далеко не всегда могут быть общеизвестными, особенно, это касается узких специальностей. Рассматриваемый метод помогает решить подобную проблему, позволяя по введённому описанию определить термин, который с наибольшей вероятностью имелся в виду.

Также он может быть внедрён в системы поиска документов по формальному описанию их содержимого.

1.3 Анализ существующих решений

Задача определения какого-либо объекта: будь то документ или термин, по его описанию ставится практически в каждой области, предполагающей повторное использование уже накопленных данных. Для автоматизации процесса поиска разрабатываются системы, ориентированные на её решение.

Autonomy

Представляет из себя поисковую систему, предоставляющую пользователям возможность делать запросы на естественном языке. Система анализирует введённый текст, извлекает из него смысловое содержание и помещает в специальный конфигурационный файл, который привлекается в дальнейшем при поиске [6].

Помимо этого, в совокупности с платформой IDOL 10 стало возможным работать практически со всеми видами представления информации: аудио, видео, электронная почта, веб-контент и структурированными машинными данными (например, журналы транзакций, показатели счётчиков и др.), позволяя извлекать смысловую информацию без предварительной обработки, особенно это касается аудио- и видеофайлов [7, 8].

Такое расширение возможностей позволило создать следующие решения:

- Autonomy Legal Hold – сбор информации для судебных разбирательств;
- Autonomy Investigator – поиск информации, связанной с фактами мошенничества;
- Autonomy Voice Discovery – работа с аудиозвонками.

Webcompass

Webcompass – поисковая система, предназначенная для специалистов (в отличие от Autonomy, которая в основном взаимодействует с конечными пользователями), которые могут структурно сформулировать свой запрос, а также промаркировать области поиска, в которых наиболее вероятно можно найти ответ [6].

И Autonomy, и Webcompass – коммерческие системы, что касается исследовательских проектов, то можно привести в пример систему MARRI.

MARRI

Поскольку большой объём информации приходится на сеть Интернет, то большинство проектов ориентировано на поиск необходимых Web-страниц. К ним относится и система MARRI, обрабатывающая запросы определённой предметной области.

Для того, чтобы решить поставленную задачу, используются знания, представленные в виде онтологий, которые в текущем проекте понимаются как множества концептов и связей между ними.

Основная идея заключается в том, что подходящие тексты содержат фрагменты, которые могут быть сопоставимы с онтологией предметной области. Таким образом, при анализе страниц проверяется их соответствие онтологическому тесту, по результатам которого система возвращает пользователю только те страницы, которые его прошли [9].

Краткий сравнительный анализ приведён в таблице 1.

Таблица 1 – Сравнительный анализ существующих решений

Решение / **Критерий**	Autonomy	Webcompass	MARRI
Возможность работы с ЕЯ	Есть	Нет	Есть
Тип обрабатываемых документов	Текстовые документы, аудио, видео, …	Web-страницы	Web-страницы
Целевая аудитория	Широкая	Узкопрофильная	Широкая
Стоимость	Высокая	Нет данных	Нет данных

1.4 Онтология

В основе множества современных проектов, ориентированных на работу с естественным языком (в том числе и тех, что были рассмотрены ранее) нередко положены онтологии.

Понятие онтологии

Термин «онтология» заимствован из философии, где под ним подразумевается система категорий, являющихся следствием определённого взгляда на мир [10].

В настоящий момент понимание этого термина различно, всё зависит от контекста и поставленной цели. Что касается информационных технологий, то

в этой сфере она рассматривается как удобная абстракция для отображения накопленных знаний в некоторой предметной области [6, 11].

Наиболее распространено следующее определение: «Онтология – это явная спецификация концептуализации» [11]. Под концептуализацией подразумевается упрощённый взгляд на мир, который нужно представить для достижения какой-либо цели. Как правило, онтологии включают не только общие, но и специфичные для рассматриваемой области термины.

Преимущество онтологий в том, что они позволяют [12]:

- накапливать знания в конкретной предметной области;
- повторно их использовать (детально проработанная онтология может быть интегрирована в несколько проектов, избавляя разработчиков от необходимости создавать её заново);
- анализировать их (анализ имеющихся терминов особо важен как при повторном использовании, так и при их расширении);
- совместно использовать накопленные знания конкретной области;
- явно выделить имеющиеся допущения.

Теоретико-модельная формализация

На рисунке 1.1 приведена общая схема моделирования системы. Как правило, моделируемая предметная область представляется в виде некоторого набора текстовых документов на естественном языке, далее на их основе строится теория предметной области, при этом особое внимание уделяется формальному описанию онтологии [13].

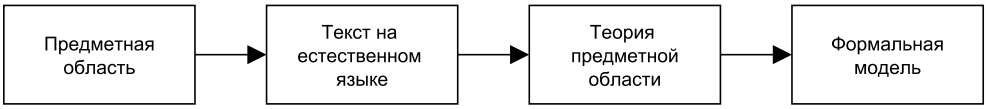


Рисунок 1.1 – Общая схема.

Модели онтологии

Под формальной моделью онтологии предметной области будем называть пару

$$O = (\sigma, A)\,, \tag{1}$$

где σ – множество ключевых понятий предметной области, A – множество предложений, описывающих их смысл.

Множество σ называют сигнатурой онтологии предметной области. Множество A состоит из определений символов, содержащихся в сигнатуре σ.

Однако множество предложений может содержать сигнатурные символы, которые не являются символами ключевых понятий предметной области. Это возникает из-за того, что при их описании использовались утверждения, в которых содержались термины другой тематики.

Онтология, которая представляется формулой (1) может быть полезна при составлении спецификаций, а именно, для описания терминов, принадлежащих узкой предметной направленности [13].

1.5 Особенности естественного языка

Естественный язык (ЕЯ) – сложная многоуровневая система, которая возникла для обмена информацией в процессе практической деятельности человека, кроме того, постоянно изменяется в связи с ней [14, 15].

Возможно несколько способов разбиения текста на ЕЯ на уровни [16]:

- синтаксический уровень (уровень предложений);
- морфологический уровень (уровень слов);
- фонологический уровень (уровень фонем для устной речи/уровень символов для письменный текстов).

Подобное разбиение условно, поскольку в зависимости от задачи может выделяться отдельно уровень морфем (значимая часть слова) как подуровень морфологического уровня. Также может быть выделен лексический уровень.

1.6 Предобработка текста на ЕЯ

Для достижения наилучшего качества обработки текста на ЕЯ необходимо сначала провести его предобработку, чтобы привести его в удобный для работы формат.

Предобработка может включать в себя [17]:

- нормализацию:
  - перевод всех букв в тексте в один регистр;
  - удаление знаков пунктуации;
  - удаление цифр/чисел или замена на текстовый эквивалент;
- токенизацию (чаще всего по словам);
- лемматизацию (приведение слова к словарной форме – лемме)/стемминг (процесс отбрасывания словообразующих морфем);
- удаление стоп-слов, т.е. слов, которые не несут смысловой нагрузки;
- векторизацию.

1.7 Векторизация

Используется для представления текста в удобном формате. Наиболее простой способ – «мешок слов» («bag-of-words») или набор ключевых слов или терминов. При таком подходе игнорируется порядок единиц, входящих в состав рассматриваемого текста.

Под терминами коллекции документов D будем понимать все одиночные слова (кроме стоп-слов), которые встретились в тексте хотя бы в одном из документов. В итоге получается множество всех терминов коллекции:

$$\tau = \{t \,|\, t - \text{ термин}\}. \tag{2}$$

Каждый документ в пространстве терминов представляется в виде вектора:

$$d = \left(t_1, ..., t_{|\tau|}\right)^T,\ d \in D, \tag{3}$$

где каждое число – координата вектора, соответствует конкретному термину и равняется его весу в данном документе.

1.7.1 Методы для вычисления веса слова в тексте

BinaryBOW

В самом простом, бинарном, случае такая координата принимает значение 1, если соответствующий термин встречается в документе, 0 – иначе [18].

CountBOW

Рассматривается следующий подход: чем чаще слово употреблено в тексте, тем больше его значимость. Таким образом, координата вектора фиксирует количество вхождений этого термина [19].

TF-IDF, Term Frequency – Inverse Document Frequency

Для того, чтобы избежать зависимости значимости термина от длины рассматриваемого документа, следует нормализовать количество его вхождений. Такая величина называется частотой термина (Term Frequency или TF).

Метод TF-IDF предполагает, что значимость термина прямо пропорциональна частоте его появления в документе и обратно пропорциональна доле документов в наборе, в которых он употреблён.

При таком подходе наибольший вес получает тот термин, который часто встречается в одном или небольшой группе документов, но не встречается в остальных, то есть, является некой отличительной особенностью на фоне часто употребляемых слов [20].

Таким образом, учитывая коллекцию документов D, термин t и текущий документ $d \in D$, вес вычисляется по формуле (4):

$$t(d) = f(t,d) \cdot \ln\left(\frac{|D|}{f(t,D)}\right), \tag{4}$$

где $f(t,d)$ – количество появлений t в d (5), $|D|$ – число документов в коллек-

ции, $f(t, D)$ – количество документов, в которых встречается рассматриваемый термин (6).

$$f(t, d) = \frac{n_t}{\sum\limits_k n_k}, \tag{5}$$

где n_i – количество появлений в рассматриваемом документе d соответствующего термина i.

$$f(t, D) = |\{d_i \in D | t_i \in d_i\}| \tag{6}$$

Может возникнуть несколько ситуаций в зависимости от принимаемых значений этими аргументами [21].

Предположим, что $|D| \sim f(t, D)$, то есть, размер корпуса документов примерно равен количеству употреблений t в D. Другими словами, термин t употребляется в большинстве документов онтологии.

Если выполняется условие (7):

$$c < \ln\left(\frac{|D|}{f(t, D)}\right) < 1, \tag{7}$$

где c – константа с малым значением, то $t(d)$ будет меньше, чем $f(t, d)$.

Это означает, что t широко распространён по всей коллекции документов. Чаще всего подобное поведение наблюдается в отношении очень общих слов, а также таких как союзы, предлоги, которые сами по себе, как правило, не несут ключевого значения. Подобные слова имеют очень низкое значение TF-IDF, тем самым, помечаются как незначительные при поиске.

С другой стороны, предположим, что $f(t, D)$ принимает очень малое значение (термин употреблён лишь в малом количестве документов онтологии), в то время как $f(t, d)$ очень большое (термин часто используется в пределах документа). Получается, что логарифм из формулы (4) также достаточно большой по величине, что напрямую сказывается на $t(d)$. Следовательно, рассматриваемый термин имеет высокий вес, что подчёркивает его важность. В таком случае

говорят, что t обладает большой дискриминационной силой.

При таком подходе необходимо учитывать тот факт, что не все «редкие» слова могут быть важны в рамках рассматриваемой задачи. Для того, чтобы сократить множество терминов, вводится понятие частоты документов или DF (Document frequency) [17].

Возможная модификация TF-IDF

Частота документов – количество документов, в которых встречается термин. Вводится пороговое значение DF, которое в отличие от удаления стоп-слов, призванного уменьшать количество высокочастотных, не имеющих отношения к теме слов, устраняет нечастые слова.

Все термины, встречающиеся меньше, чем в m документах коллекции, не рассматриваются, где m – заранее определённый порог.

Пороговое значение DF основано на предположении, что нечастые слова не являются информативными. Если установить его в 1, то термины, которые встречаются только в одном документе, учитываться не будут.

Адаптация TF-IDF к рассматриваемой задаче

Разрабатываемый метод строится на предположении, что, несмотря на то, что определения дают разные люди, они пересекаются в ключевых моментах. Иными словами, есть какие-либо свойства, характеристики, которые являются отличительной чертой рассматриваемого объекта и о которых опрашиваемые с наибольшей вероятностью укажут.

Основываясь на этом, ключевые слова – это те, которые употребляются большинством, и, как правило, свидетельствуют о признаках, присущих определяемому термину, и должны иметь наибольший вес.

В TF-IDF слова с подобной частотой рассматриваются как слишком общие, не несущие информационную нагрузку, а в основе разрабатываемого метода лежит противоположное предположение. Для того, чтобы адаптировать

упомянутый подход к решению поставленной задачи, необходимо модифицировать формулу. Так, значимость слова следует вычислять по формуле 8:

$$t(d) = f(t, d) \cdot \ln\left(\frac{|D|}{|D| - f(t, D)}\right). \quad (8)$$

Для того, чтобы всё-таки отсеять общие слова с минимальной информационной нагрузкой, следует ввести пороговые значения: если слово употребляется меньше, чем в 10% определений, скорее всего оно никак не характеризует термин, и если больше 90%, считается, что слово не является ключевым.

1.8 Поиск нечётких дубликатов

1.8.1 Общие понятия

Будем считать два объекта дубликатами, если они полностью совпадают. Если же один из них представляет из себя видоизменённую копию другого, в таком случае, они являются нечёткими дубликатами.

В качестве объекта может приниматься текст, изображение и т.п. В рамках поставленной задачи будет рассматриваться обработка текста.

В основном, алгоритмы поиска нечётких дубликатов основываются на создании либо сигнатуры объекта и её поиск в уже имеющейся базе сигнатур, либо коллекции из слов рассматриваемого документа и сравнения с заранее составленными коллекциями [22].

Существует несколько алгоритмов определения дубликатов, среди них метод шинглов и косинусное сходство.

1.8.2 Метод шинглов

Шингл – небольшой, состоящий из нескольких слов, фрагмент текста [23]. Количество слов M называется длиной шингла и подбирается в зависимости от задачи. Документ разбивается на шинглы, которые создаются, как правило, внахлёст.

Для достижения наилучшего результата рекомендуется производить предобработку текста (раздел 1.5) и выбор ключевых терминов (раздел 1.7.1), как было описано ранее.

При таком подходе, в составе шинглов, в основном находятся наиболее важные слова, что увеличивает шанс обнаружения нечётких дубликатов.

Заранее отсортировав полученное множество по алфавиту и сформировав блоки по M элементов, к каждому применяется хеш-функция [24].

Таким образом, имеет место соответствие: шингл – число, по которому потом будет производиться сравнение между шинглами рассматриваемого текста и уже имеющимися.

Выделяются две меры, по которым можно судить о схожести сравниваемых единиц: мера сходства (9) и мера вхождения (10) [25].

$$\text{rs}(A, B) = \frac{|A \cap B|}{|A \cup B|}, \tag{9}$$

$$\text{ent}(A, B) = \frac{|A \cap B|}{|A|}, \tag{10}$$

где A – множество терминов первого текста; B – множество терминов второго текста.

Этот метод позволяет находить совпадающую информацию целыми блоками, с другой стороны, с увеличением размера шингла становится затруднительным обнаружение совпадения сочетаний из малого количества слов.

1.8.3 Векторная модель

При таком подходе между двух векторов, сформированных так, как описано в разделе 1.7, определяется мера сходства, которая называется косинусной [26].

Пусть есть вектор запроса $\mathbf{q}$ и вектор i-ого документа $\mathbf{d}_i$ (рисунок 1.2).

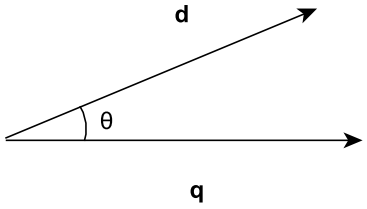


Рисунок 1.2 – Косинусное сходство.

Косинусное сходство вычисляется по формуле (11):

$$k_i = \frac{(\mathbf{q},\, \mathbf{d}_i)}{|\mathbf{q}|\,|\mathbf{d}_i|}. \tag{11}$$

Соответственно, чем значение ближе к 1, тем угол между векторами ближе к 0 градусам и два рассматриваемых вектора более схожи.

Таким образом, формируется множество $K = \left\{k_1, ..., k_{|\tau|}\right\}$. Наиболее подходящим под исходное описание считается тот термин, косинусное сходство которого является наибольшим в полученном множестве K.

Поскольку в рассматриваемой задаче все элементы любого из векторов являются неотрицательными (т.к. обозначают вес), то если $k_i = 0$, где $i = \overline{1, |\tau|}$, это означает, что термины запроса отсутствуют в рассматриваемом документе.

В отличие от метода шинглов такой подход не рассматривает информацию блоками, что, с одной стороны, лишает возможности проверки одновременного присутствия конкретных слов, но, с другой стороны, позволяет найти неточные совпадения лишь по части некоторых из них.

1.9 Семантические сети

Кроме традиционного подхода в виде «мешка слов» для представления знаний используются семантические сети, которые строятся на графах.

Семантическая сеть – ориентированный граф, в котором вершины соответствуют конкретным фактам, общим понятиям, объектам, а дуги – отношениям или ассоциациям между ними [27].

Ассоциативному подходу уделяется особое внимание в силу того, что он описывает рассматриваемый объект в терминах его связей (по-другому, ассоциаций) с другими объектами. Ассоциации определяют, прежде всего, его свойства и поведение.

Объект может быть представлен как совокупность (рисунок 1.3), состоящая из:

- характеристик и свойств;
- действий;
- набора состояний;
- множества объектов, так или иначе связанных с ним.

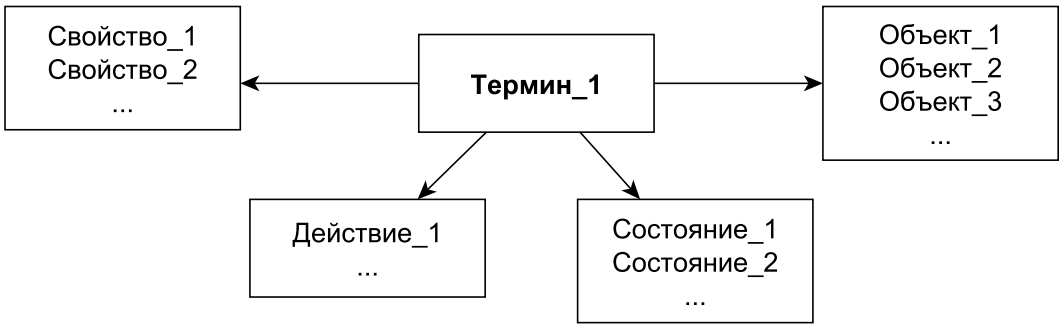


Рисунок 1.3 – Представление объекта при ассоциативном подходе.

Сеть строится из графов, и в случае совпадения узлов двух или более графов, они должны быть объединены и обновлены в соответствии с информацией, которая содержалась в узлах до их соединения. Пример простейшей сети представлен на рисунке 1.4.

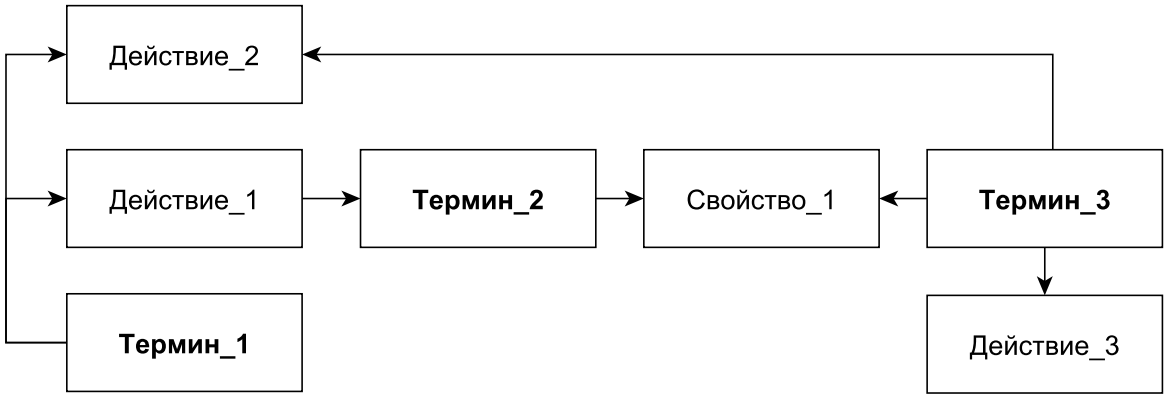


Рисунок 1.4 – Простейшая семантическая сеть.

Термины 2 и 3 обладают одинаковым признаком «Свойство 1», поэтому оба указывают на один узел, аналогичная ситуация наблюдается с «Действием 2».

1.10 Синтаксическое дерево

Применимо к решаемой задаче, каждое предложение из онтологии может быть представлено в виде синтаксических деревьев, демонстрирующих связи между словами. Синтаксические единицы, как правило, изображаются в виде узлов, а связи – дуг.

В синтаксическом дереве существует единственный узел, в который не входит ни одна дуга, и называется вершиной или корнем. Изображается он всегда сверху. Важной особенностью подобных структур является то, что в каждый узел (кроме корня) входит ровно одна дуга.

В традиционной грамматике русского языка в качестве вершины предложения рассматривается подлежащее, в то время как в современной лингвистике корнем, как правило, считается сказуемое, от него напрямую или косвенно зависят все остальные члены предложения. Такой подход предложил французский лингвист Люсьен Теньер [28].

Согласно Теньеру, предложение – «драма в миниатюре», в центре которой находится действие. Вершина, глагол-сказуемое, называется предикатом,

все остальные зависимые слова – актанты (к ним относятся подлежащее, дополнения) и сирконстанты (обстоятельства).

Одному предложению соответствует только одно синтаксическое дерево.

В Национальном корпусе русского языка [29] представлен синтаксический корпус объёмом около 1 миллиона примеров, оснащённых лингвистической разметкой, для каждого построено синтаксическое дерево, следуя принципам Теньера.

На рисунке 1.5 представлено одно из них для предложения: «Спрос на рынке труда постоянно меняется.» [30].

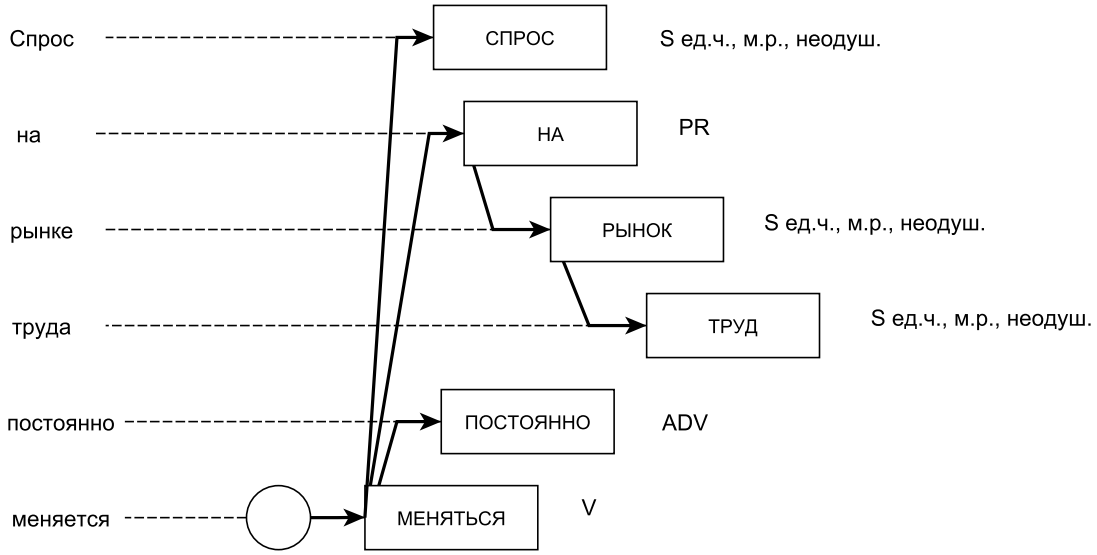


Рисунок 1.5 – Пример синтаксического дерева.

1.11 Синтаксический граф

Нередки случаи, когда одно и тоже слово в предложении повторяется, следовательно, в синтаксическом дереве будет создано несколько узлов, отведённых под один термин, но имеющих разных родителей.

Во избежании дублирования информации и для упрощения процедуры составления семантической сети, лучше создавать по одному узлу на термин, но предусматривать возможность указания ему нескольких родителей. Из-за этого, в структуре данных могут возникать циклы. Следовательно, от формулировки «синтаксическое дерево» следует перейти к «синтаксическому графу», так как

первое не допускает наличие циклов.

Выводы

Таким образом, в данной работе будет решаться задача определения объекта из ограниченной выборки терминов, касающихся двигателя внутреннего сгорания по его нечёткому описанию на русском языке.

Были проанализированы существующие методы решения, выявлены основные преимущества и недостатки, среди которых невозможность некоторых из них работать с естественным языком и высокая стоимость.

Также проанализированы основные алгоритмы работы с естественным языком, применимые к данной задаче. Был предложен способ её решения с помощью комбинации таких методов, как TF-IDF для формирования онтологии и сеть синтаксических графов в качестве вспомогательного метода, который будет привлекаться только в случае, если результаты обработки первым методом не будут удовлетворять критерию для принятия решения (степень уверенности меньше 50%).

Также для поиска нечётких дубликатов между запросом пользователя и собранными заранее данными предлагается привлечение косинусного сходства.

2 Конструкторская часть

2.1 Формат входных данных

В качестве входных данных выступает пользовательское описание какого-либо объекта. Оно должно соответствовать следующим требованиям:

1) определение даётся полностью на русском языке;
2) недопустимо использование аббревиатур, сокращений и т.д., все слова должны быть употреблены в полной форме;
3) описание должно укладываться в 1-2 предложения;
4) следует связно излагать свои мысли;
5) допускается голосовой ввод, результат которого в дальнейшем будет преобразован в текстовый формат.

2.2 Формат выходных данных

Выходные данные представляются как множество терминов из выборки, каждому из которых поставлены в соответствие величина косинусного сходства и её промасштабированное значение, выраженное в процентах.

Если в ходе работы метода дополнительно привлекалась сеть синтаксических графов, то пользователю также предоставляется информация о количестве совпавших слов (в запросе пользователя и терминах сети) и процентное соотношение.

Для интерпретации полученных значений косинусного сходства и количества совпавших слов используется функция softmax [31] позволяющая перевести множество полученных значений в вектор процентов уверенности в том, что был описан соответствующий термин.

2.3 IDEF0

Разрабатываемый метод состоит из нескольких этапов, которые представлены на рисунках 2.1-2.3. Необходимо по изложению пользователя определить

описываемый объект.

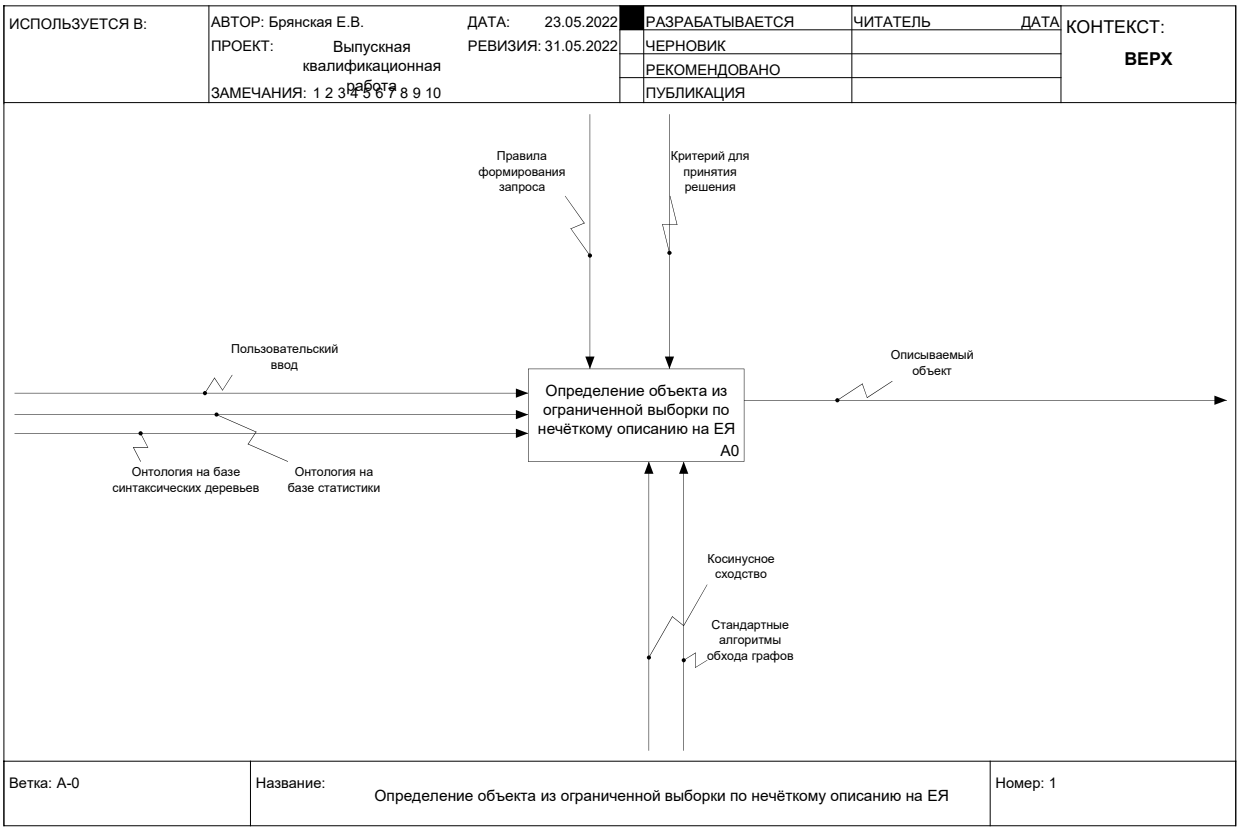


Рисунок 2.1 – Контекстная диаграмма основного метода.

Эта задача решается в несколько этапов: обработка запроса клиента, его преобразование, путём извлечения ключевых слов, сопоставление полученных данных с уже имеющимися онтологиями (сформированными на базе статистики и синтаксических графов), формирование промежуточного результата, и затем – принятие решения о том, какой именно термин (один или несколько) наиболее подходит под это описание.

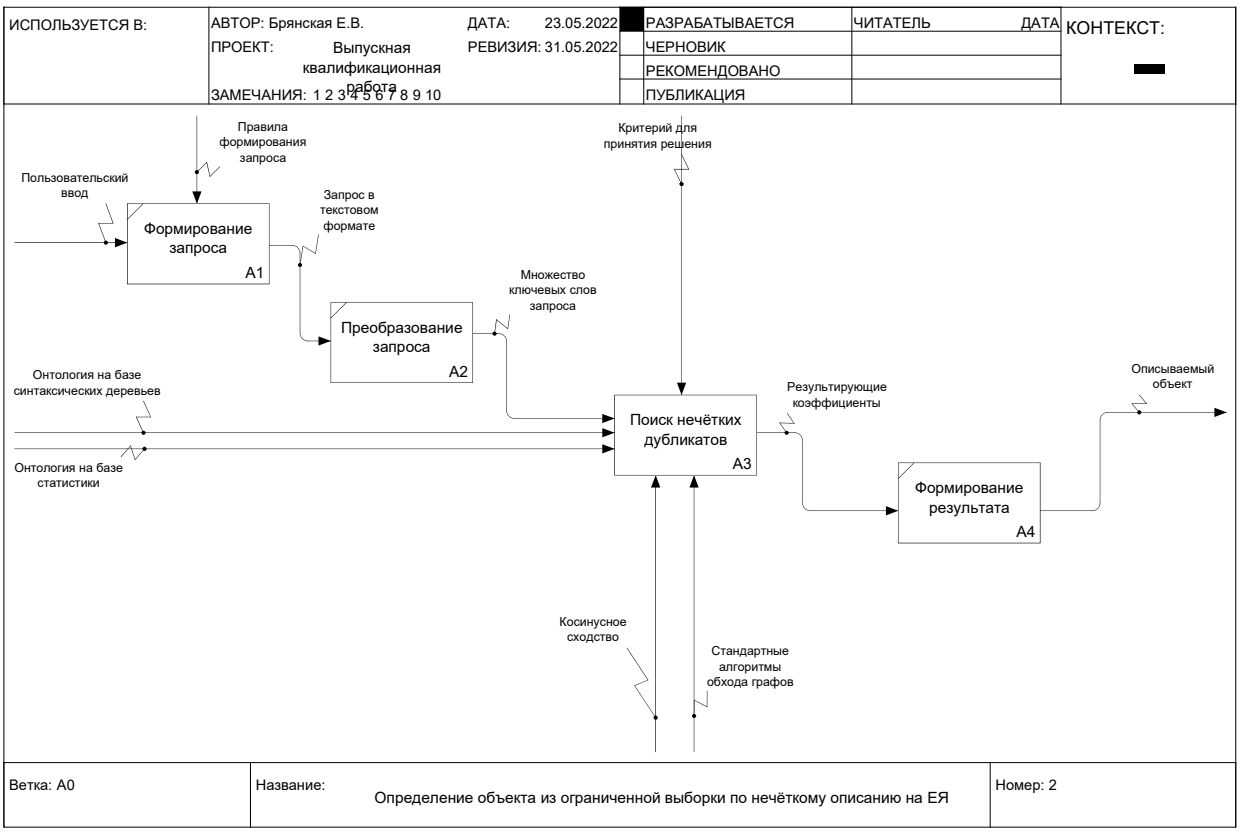


Рисунок 2.2 – Декомпозиция основного метода.

Алгоритм поиска нечётких дубликатов также состоит из нескольких этапов: сначала с помощью косинусного сходства определяются косинусные расстояния запроса пользователя и всех элементов статистической онтологии, затем, на основе полученных результатов принимается решение о том, нужно ли привлекать онтологию на основе синтаксических графов, чтобы увеличить точность распознавания термина.

Если же результаты, полученные на первом этапе, удовлетворяют критерию принятия решения, то дальнейший анализ производиться не будет.

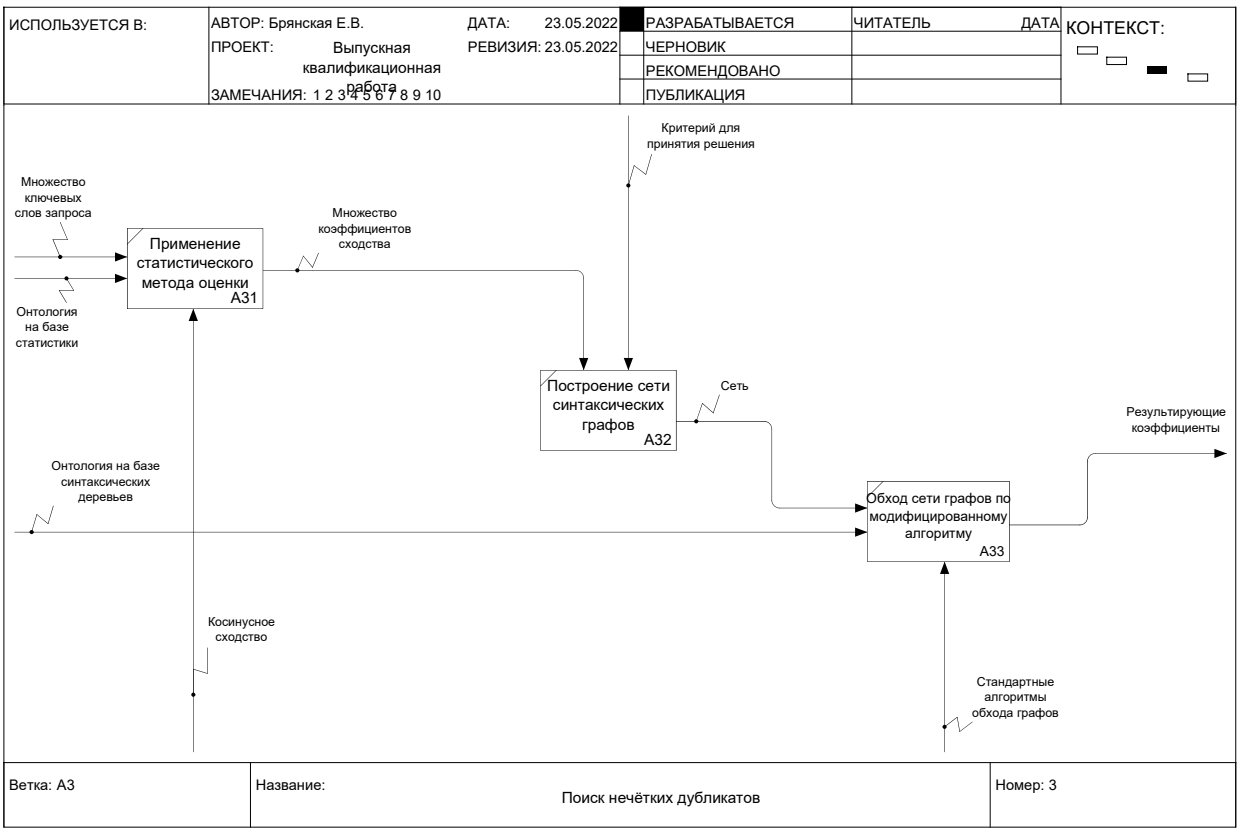


Рисунок 2.3 – Декомпозиция блока A3.

В методе используется два подхода: статистический и на основе семантической сети, для каждого из них формируется онтология, процесс создания которой состоит из нескольких последовательных шагов.

Так формирование онтологии, в основе которой лежит статистические данные, представлено на рисунках 2.4-2.5.

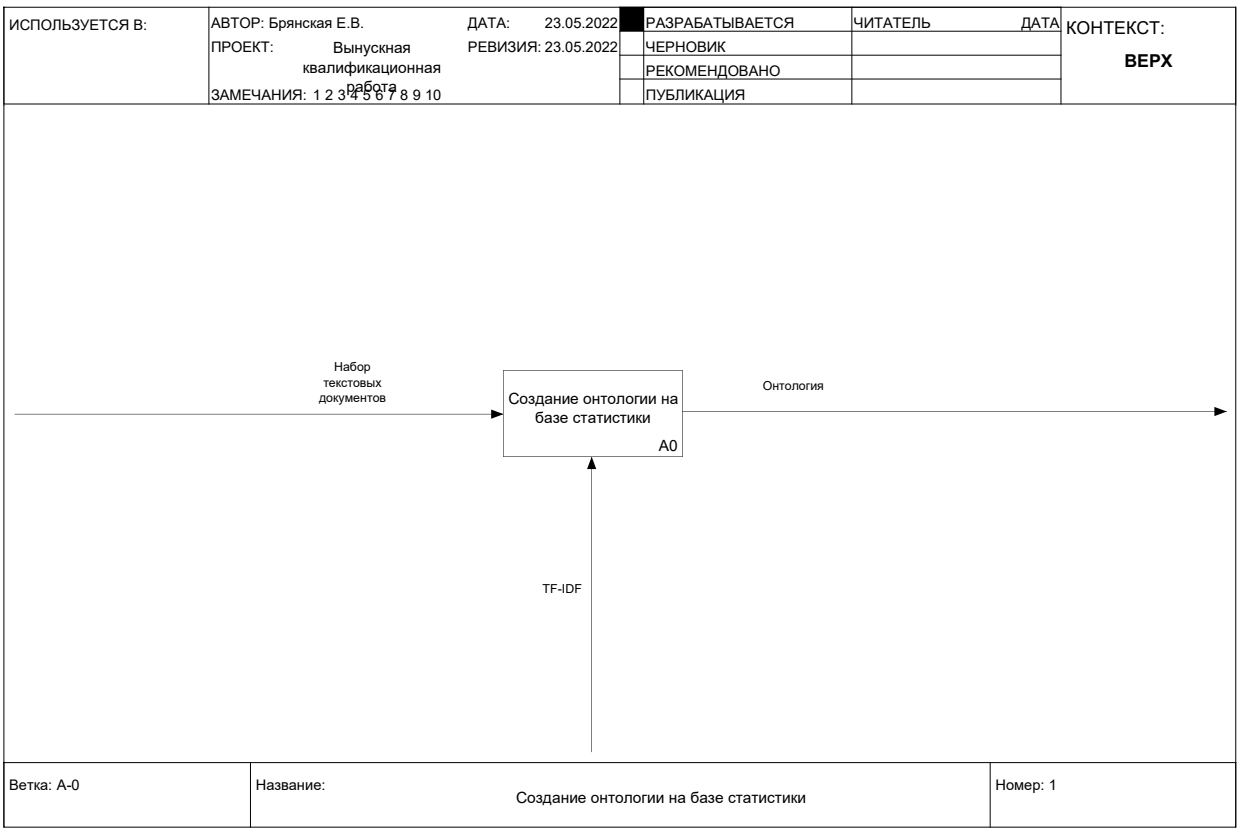


Рисунок 2.4 – Контекстная диаграмма метода создания онтологии на основе статистики.

Сначала отбираются документы, содержащие определения одного термина, далее информация в каждом из них предобрабатывается, как описано в разделе 1.6, и представляется в виде вектора.

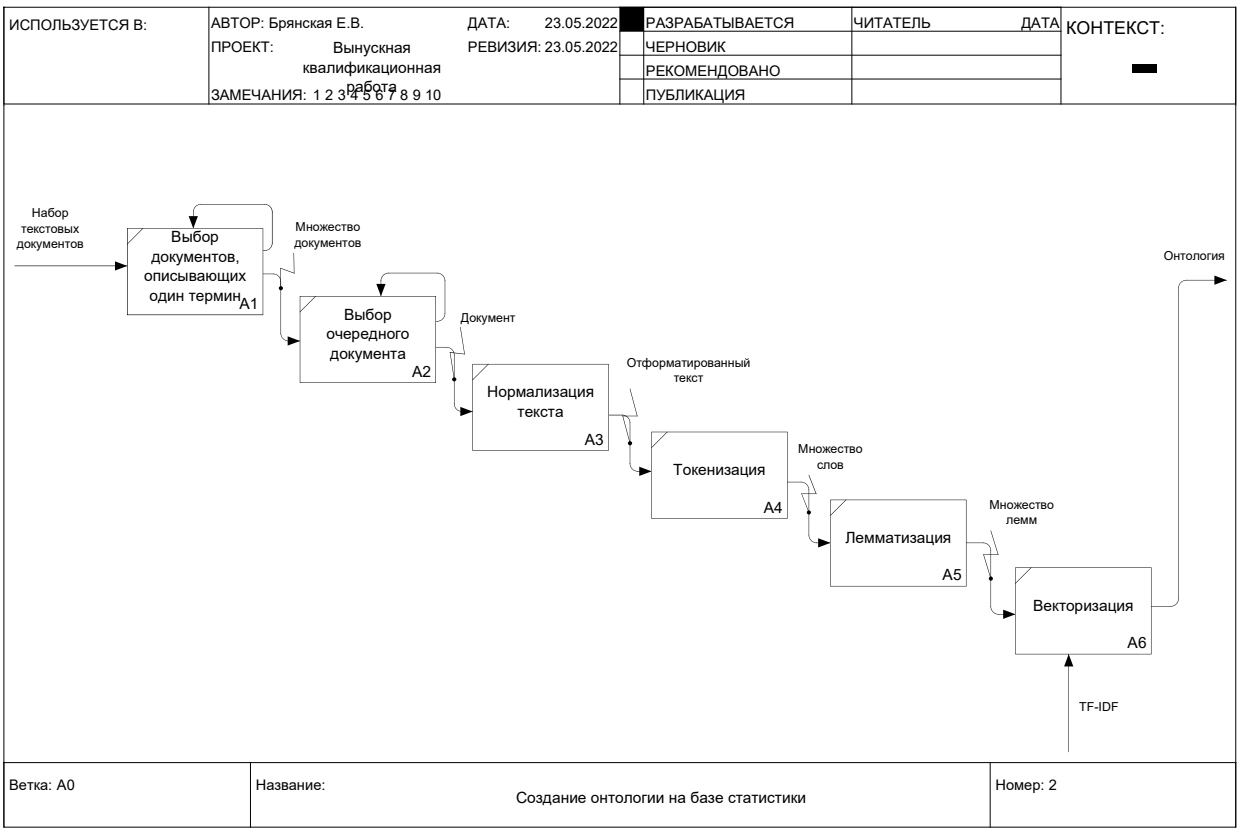


Рисунок 2.5 – Детализация метода создания онтологии на основе статистики.

Шаги создания онтологии на основе синтаксических графов представлены на рисунках 2.6-2.8.

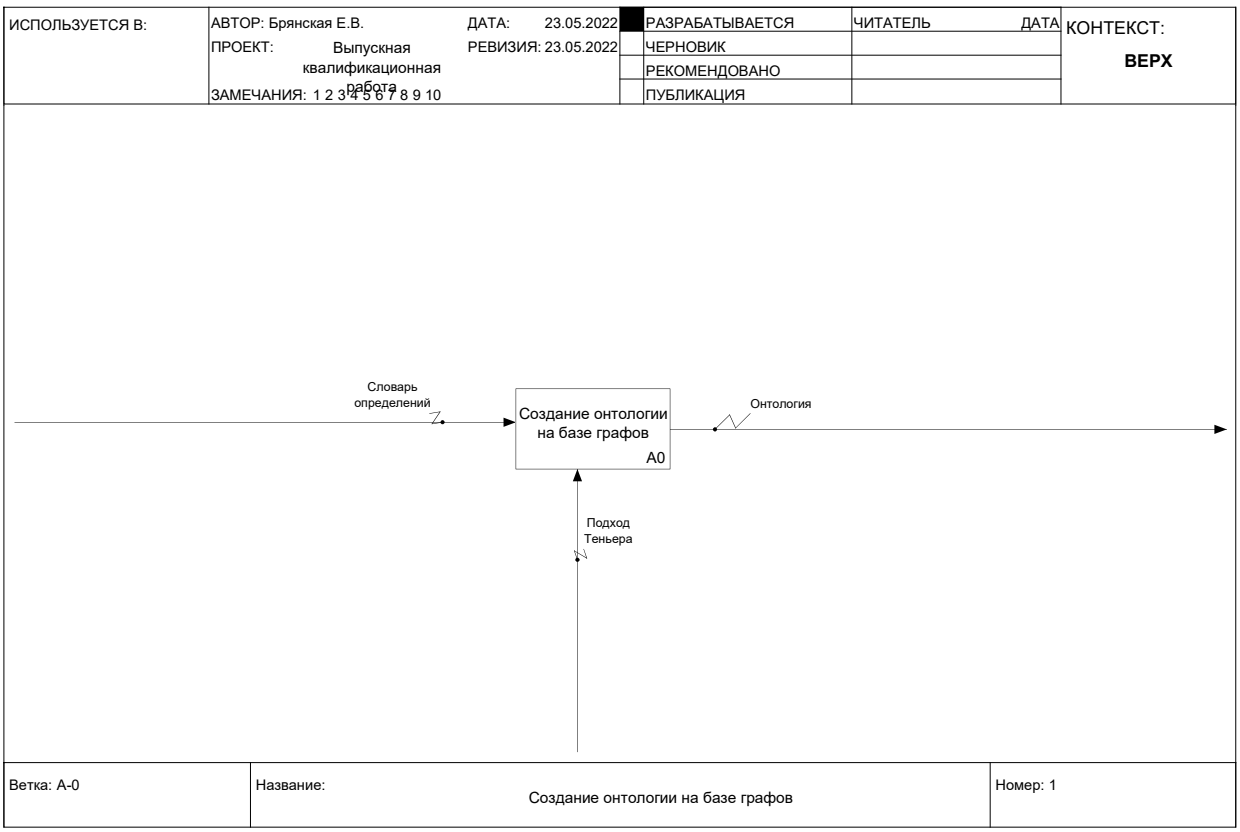


Рисунок 2.6 – Контекстная диаграмма метода создания онтологии на основе графов.

Каждый термин онтологии описывается ровно одним предложением, и для каждого из них строится соответствующий граф. Для всех слов в предложении определяются постоянные и непостоянные признаки, в предложении выделяются словосочетания, и на основе полученных данных формируется граф, который и становится элементом онтологии.

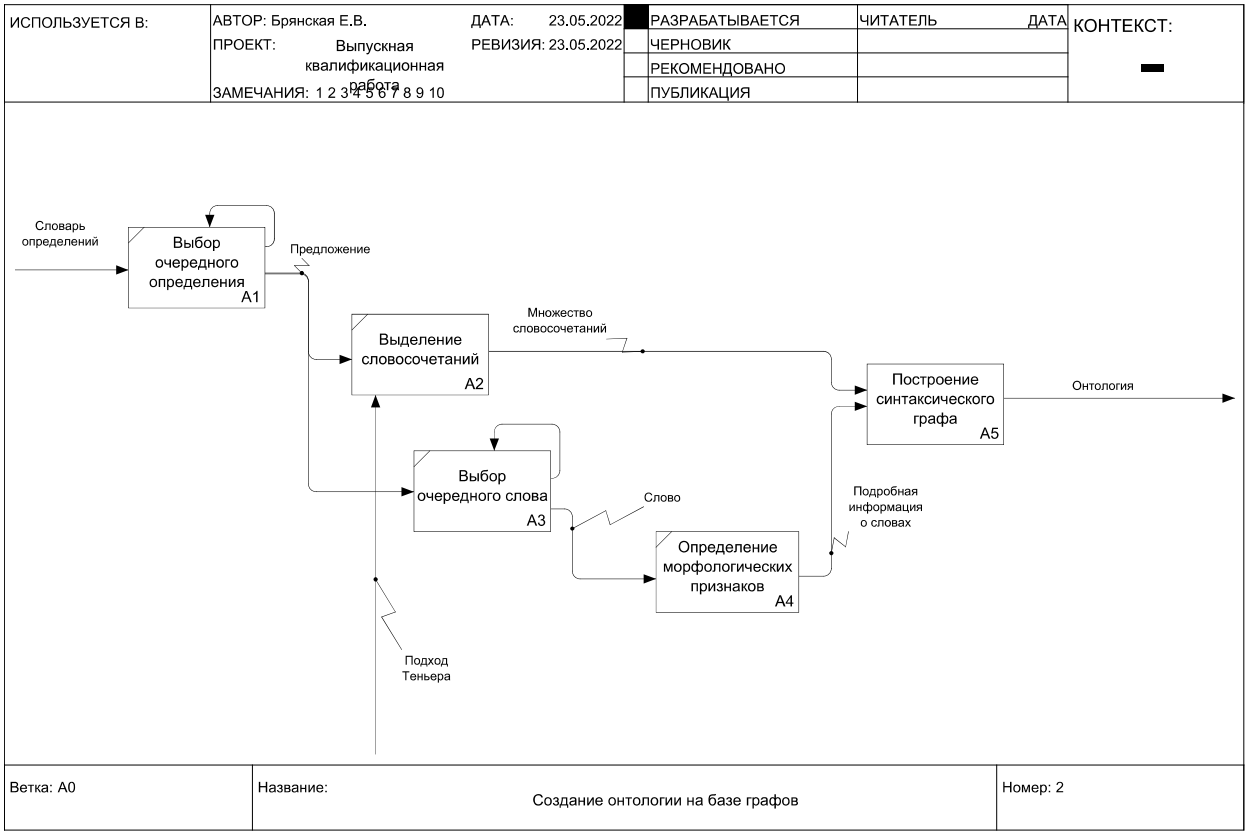


Рисунок 2.7 – Детализация метода создания онтологии на основе графов.

Построение графа начинается с инициализации корня, затем последовательно обрабатывается каждое словосочетание и на основе полученной морфологической информации осуществляются дальнейшие действия: узел может быть добавлен или нет.

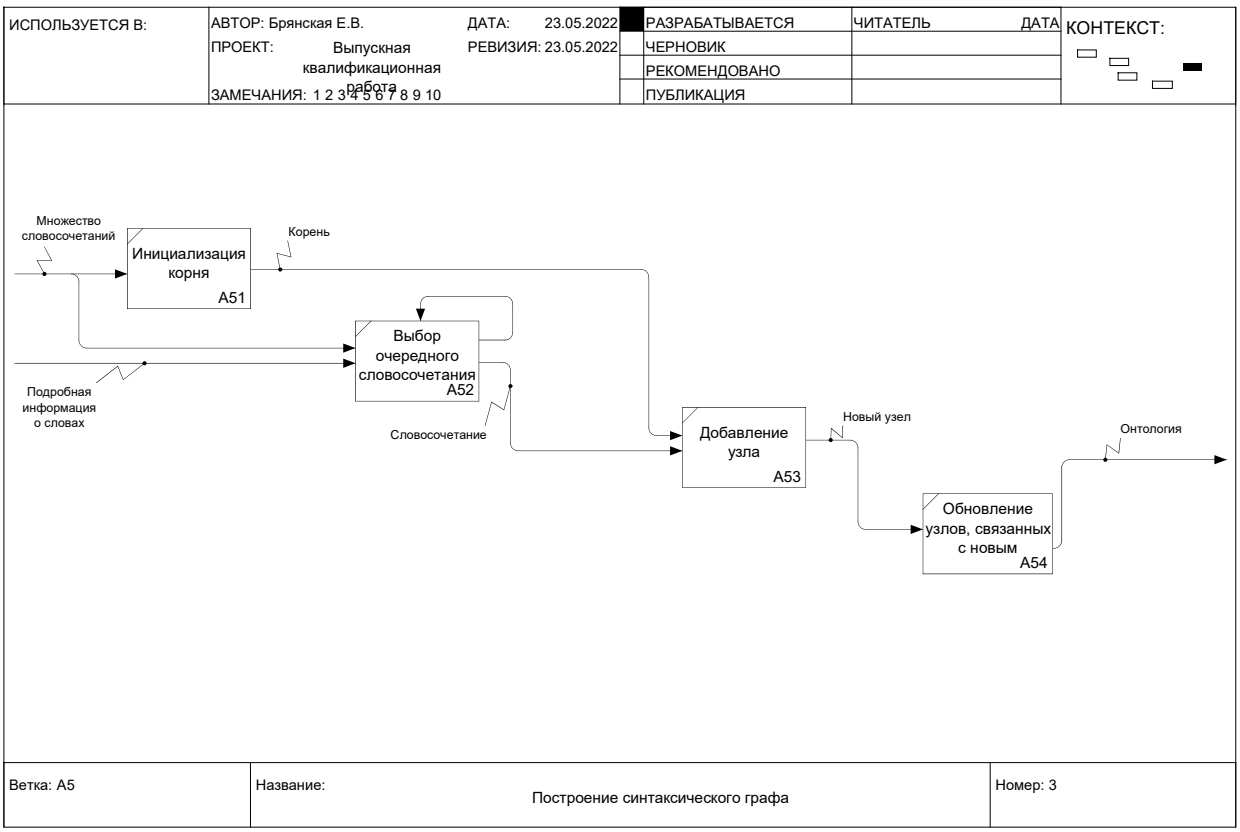


Рисунок 2.8 – Декомпозиция блока A5.

2.4 Ключевые алгоритмы

2.4.1 Основной алгоритм

Можно выделить следующие основные этапы:

1) предобработка пользовательского ввода;
2) выделение ключевых слов в запросе;
3) поиск нечётких дубликатов, с использованием статистического метода;
4) поиск нечётких дубликатов, с использованием сети из синтаксических графов (в случае не соответствия критерию принятия решения).

Схема алгоритма приведена на рисунке 2.9. Все шаги более детально разобраны далее.

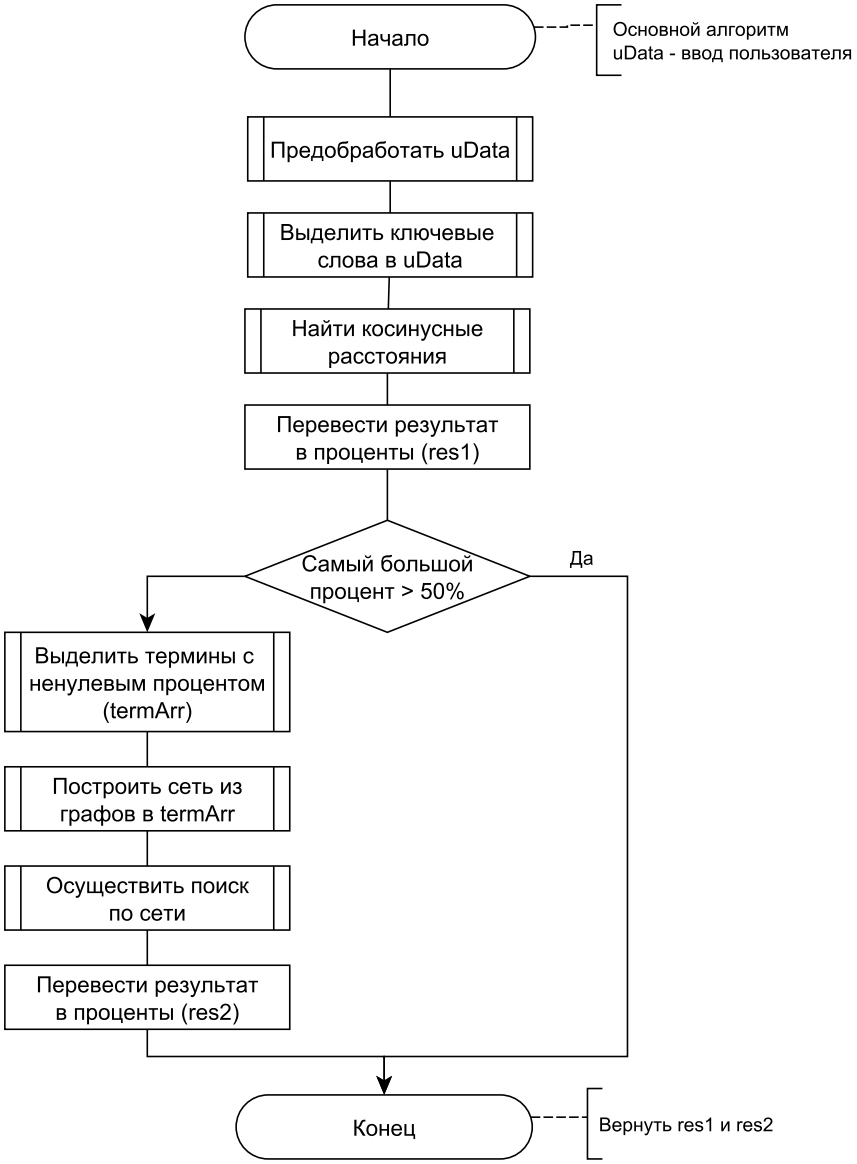


Рисунок 2.9 – Схема основного алгоритма.

2.4.2 Алгоритм предобработки данных

Предобработке подвергаются:

- пользовательский ввод (в случае голосового ввода, он сначала переводится в текст, затем уже он подвергается обработке);
- текстовые данные для формирования онтологии на базе статистических данных;
- определения из словаря, предназначенные для создания синтаксических графов.

На рисунке 2.10 представлена схема алгоритма обработки данных.

В процессе нормализации всё переводится в нижний регистр, удаляются все символы, кроме, букв русского языка, между словами выставляется фиксированно только один пробел.

Токенизация – разделение на слова. На этапе лемматизации все слова приводятся к начальной форме, затем некоторые из них будут удалены, как стоп-слова. И для удобства распознавания терминов буква «ё» заменяется на «е».

Рисунок 2.10 – Схема алгоритма предобработки данных.

2.4.3 Алгоритм создания онтологии на базе статистических данных

На рисунке 2.11 изображена схема алгоритма поиска ключевых слов. В основе лежит метод TF-IDF, описанный в разделе 1.7.1.

Каждое описание проходит предобработку по алгоритму, который представлен в предыдущем разделе, далее для каждого слова подсчитывается число документов, в которых оно употреблено, и сколько раз.

Затем вычисляется сам вес слова, как произведение этих величин, и в случае, если он удовлетворяет заранее заданным границам, то это слово считается

ключевым для рассматриваемого термина. Подобные слова и формируют «признаки», по которым в дальнейшем и будет проходить идентификация объекта.

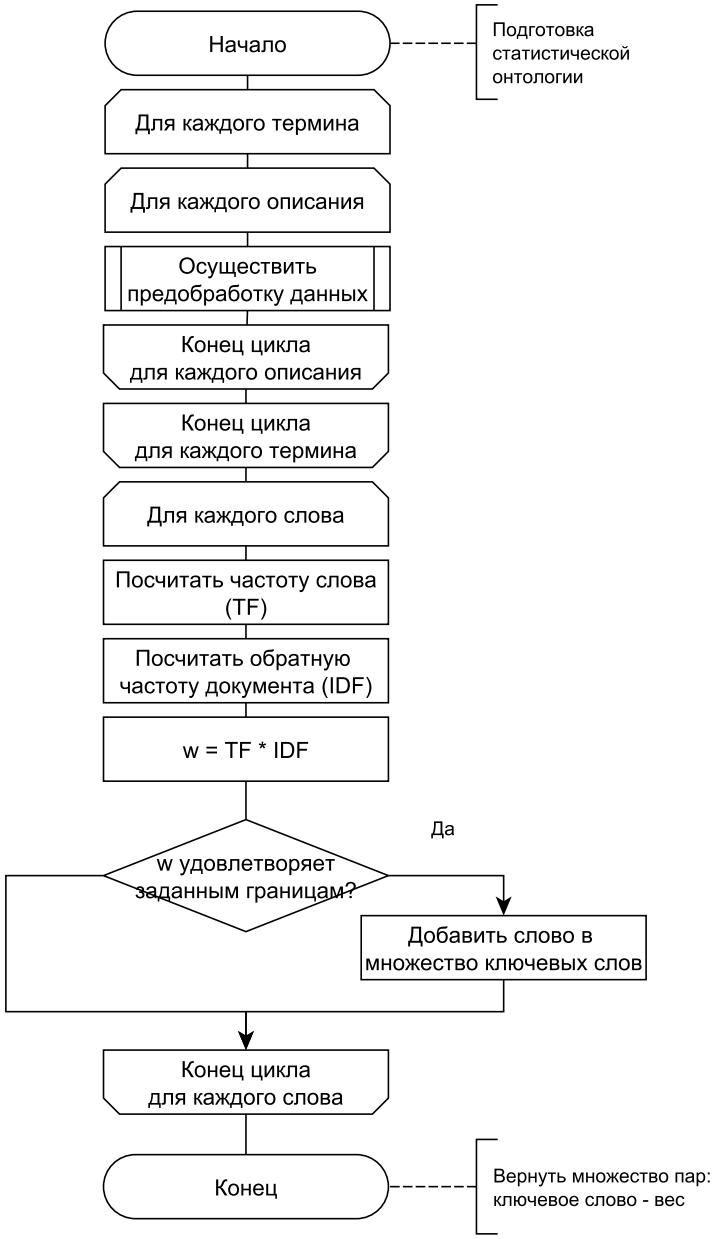


Рисунок 2.11 – Схема алгоритма создания онтологии с использованием статистического метода.

2.4.4 Алгоритм создания онтологии на основе синтаксических графов

Из-за того, что в большинстве инструментов для определения словосочетаний заложены принципы, которые определил Теньер, для решения постав-

ленной задачи необходимо идентифицировать служебные части речи.

В случае обнаружения и наличия зависимых от них слов, менять у детей этого элемента идентификатор родителя на идентификатор родителя для обрабатываемого слова. Таким образом, в синтаксическом дереве будут только слова, которые так или иначе несут смысловую нагрузку.

На рисунке 2.12 приведена подробная схема алгоритма.

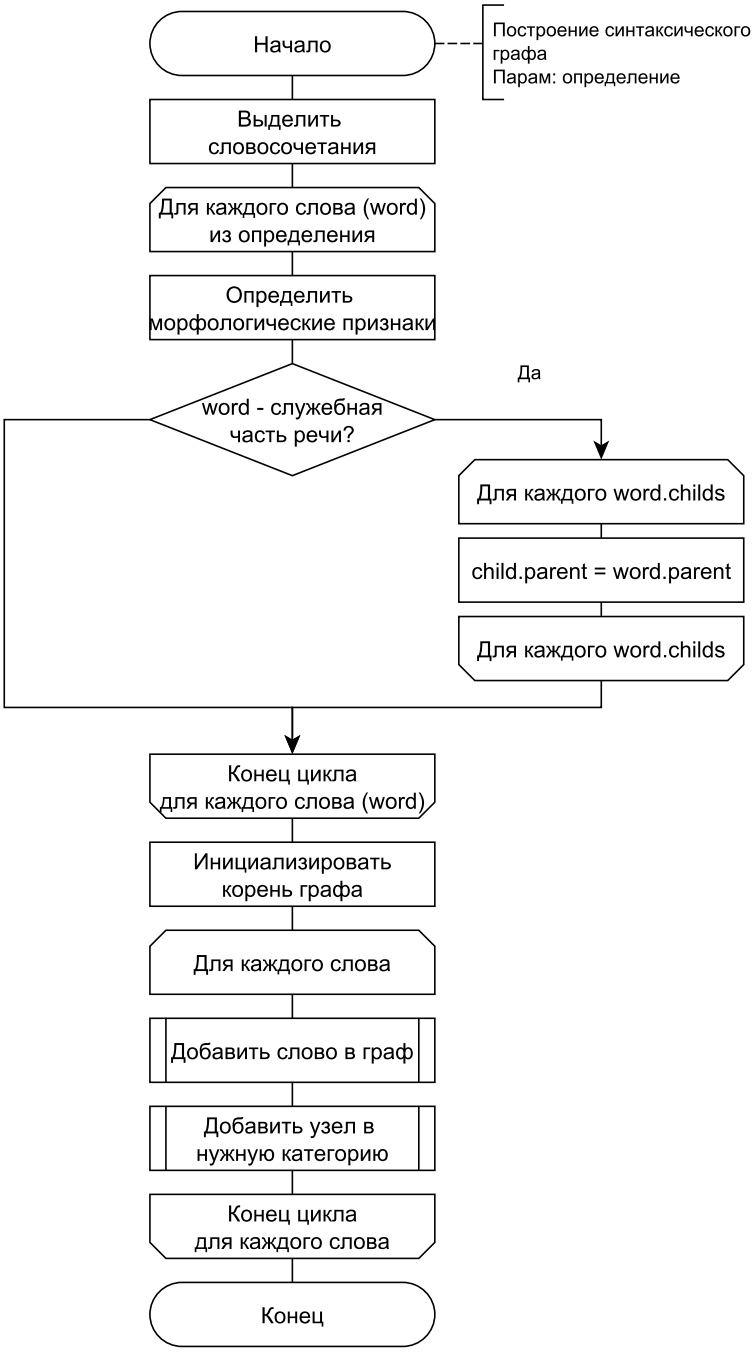


Рисунок 2.12 – Схема алгоритма создания онтологии на основе графов.

Само добавление узла в граф (рисунок 2.13) осуществляется только тогда, когда было установлено наличие родителя в графе. В противном случае создаётся «пустой» узел с идентификатором родителя.

Если на момент добавления, в графе нет узла, ассоциированного с рассматриваемым словом, то добавляется новый узел. Если же узел есть, то он дополняется информацией (идентификаторы родителей, группы категорий и т.д.).

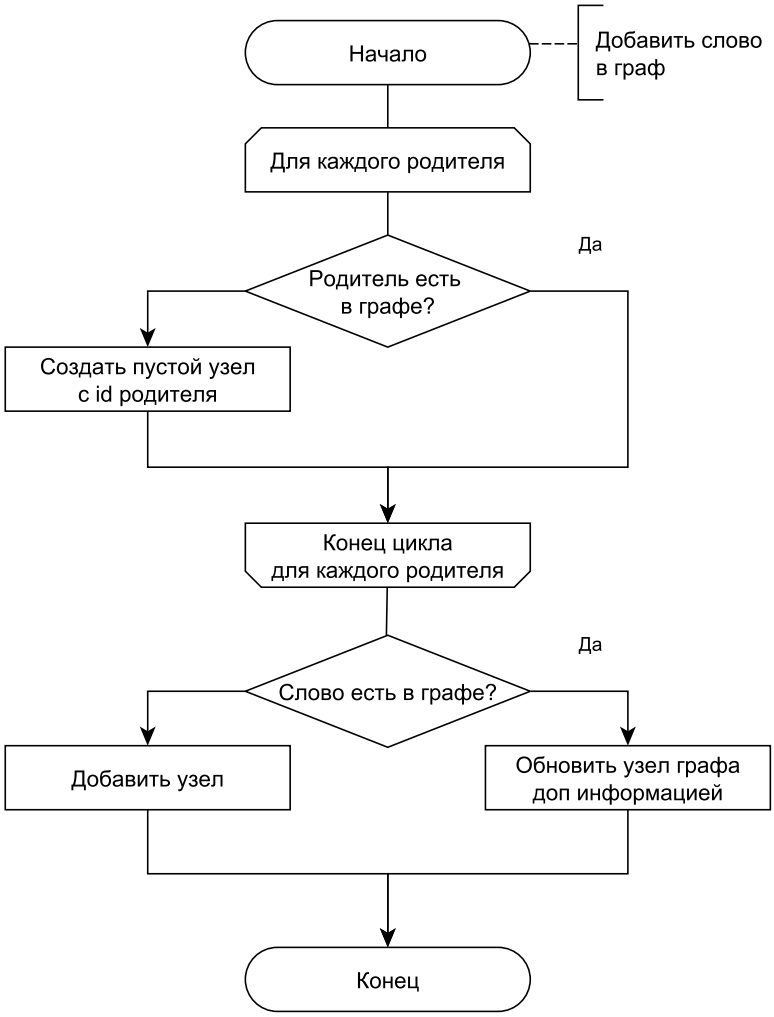


Рисунок 2.13 – Схема алгоритма добавления слова в граф.

Поскольку при идентификации объекта по его определению сначала просматриваются все объекты, связанные с термином, затем действия и т.д., то необходимо при формировании графа распределять детей каждого из узлов по категориям (рисунок 2.14). Это возможно благодаря заранее определённому по-

стоянному морфологическому признаку – часть речи.

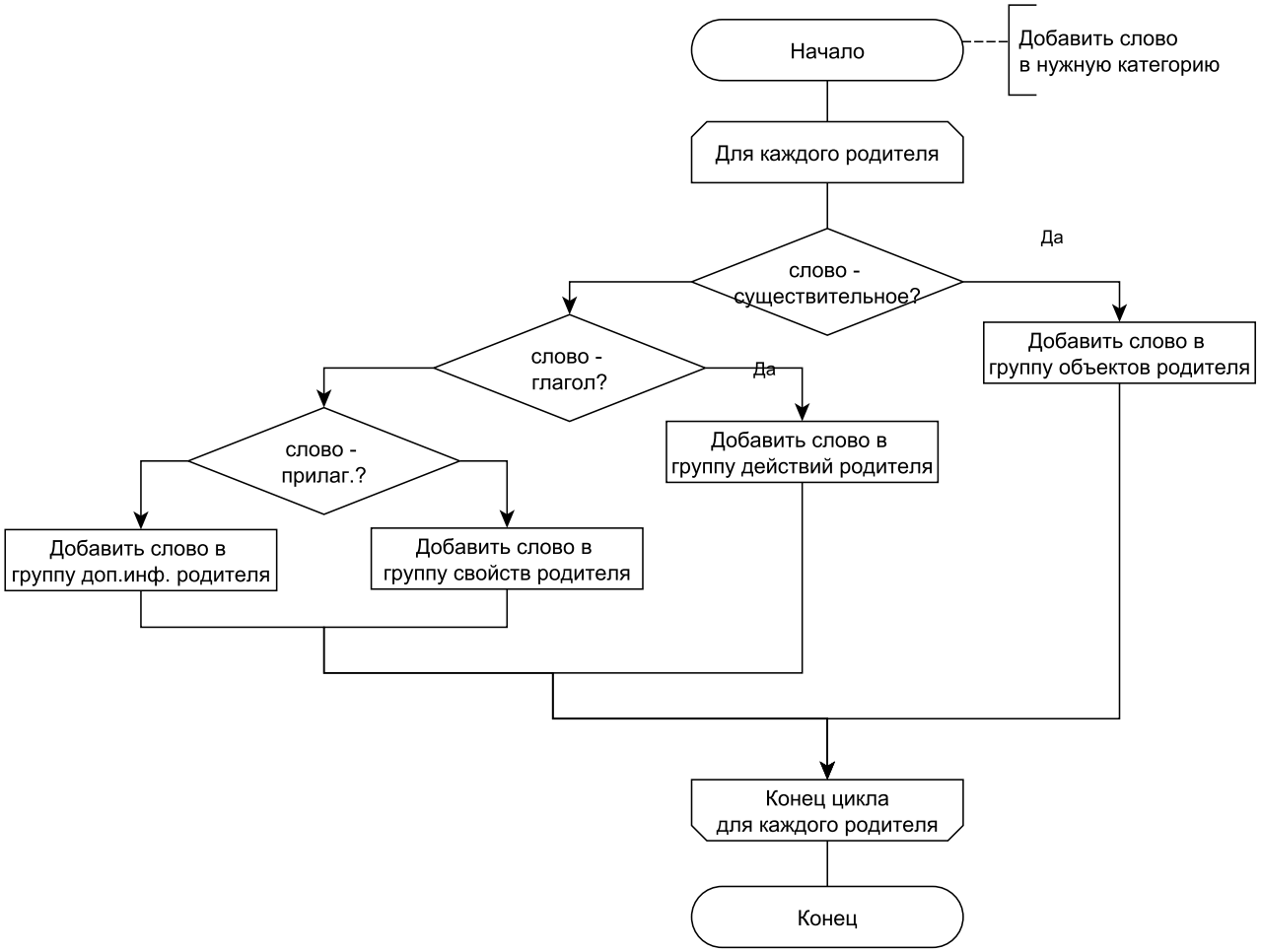


Рисунок 2.14 – Схема алгоритма добавления слова в нужную категорию.

2.4.5 Алгоритм построения сети

Сеть строится из синтаксических графов. При их слиянии необходимо следить за тем, чтобы не было дублирующих узлов (то есть, узлов, ассоциированных с одним и тем же словом).

При обнаружении повторяющихся единиц, необходимо слить полезную информацию в уже существующий в графе узел.

На рисунке 2.15 подробно изложен алгоритм построения сети.

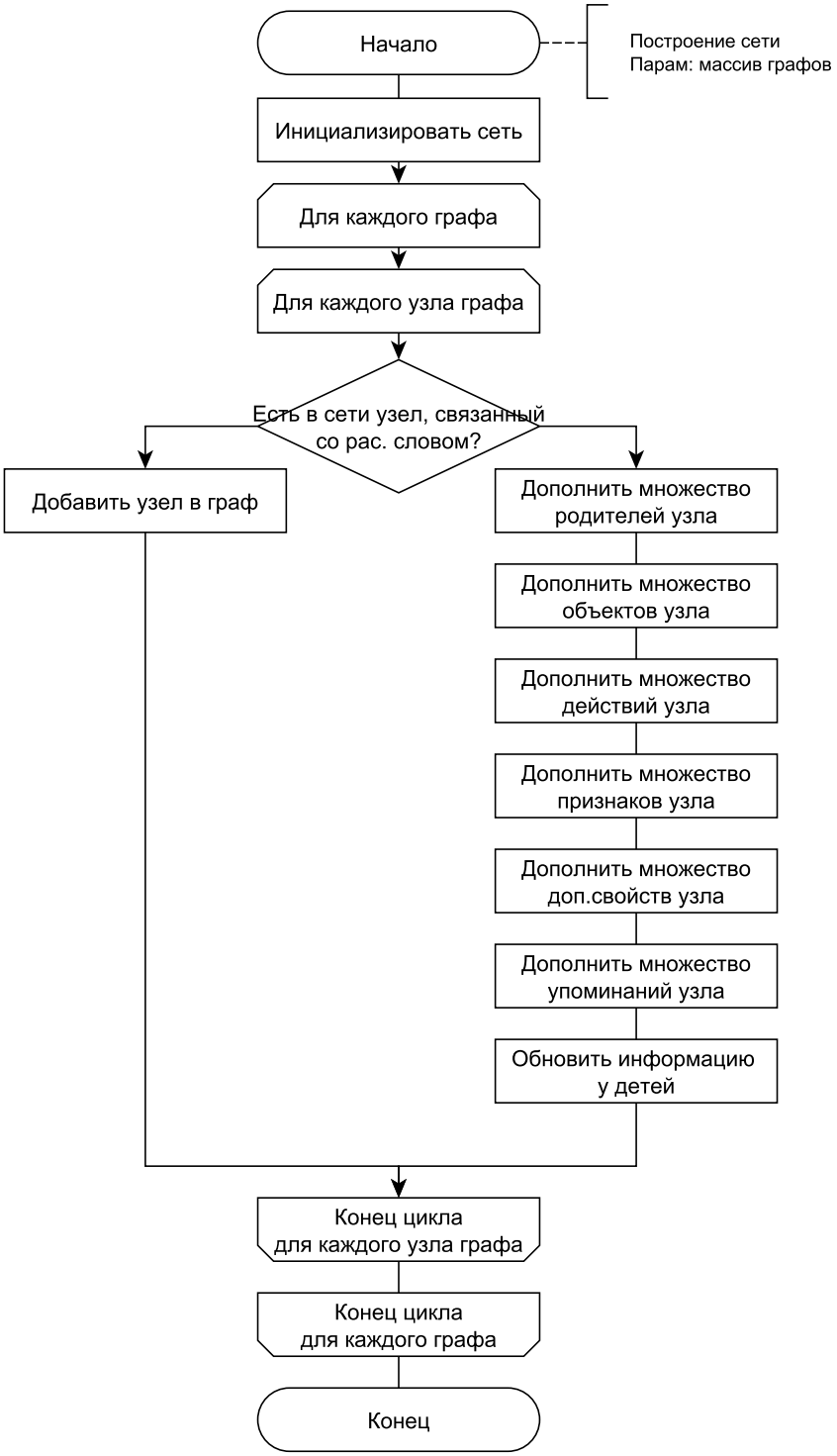


Рисунок 2.15 – Схема алгоритма построения сети.

2.4.6 Алгоритм поиска косинусного сходства

Для того, чтобы найти нечёткие дубликаты, необходимо найти косинусное расстояние между тем, что ввёл пользователь, и заранее составленной онтологией.

На этом этапе привлекается формула 11. Каждое вычисленное значение сохраняется для дальнейшего принятия решения, какие термины наиболее подходят под описание пользователя.

На рисунке 2.16 представлена схема алгоритма. Результатом его работы является массив косинусных расстояний.

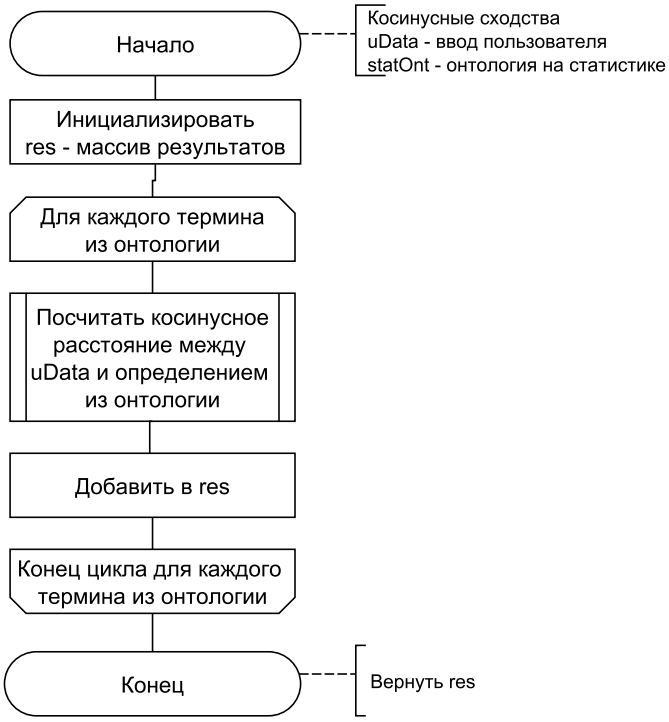


Рисунок 2.16 – Схема алгоритма поиска косинусного сходства.

2.4.7 Алгоритм поиска по сети

В качестве структуры данных используется дек, который позволяет добавлять элементы как в «голову», так и в «хвост».

По умолчанию сеть обходится в ширину (элемент снимается с головы, добавляется в хвост), причём каждое слово, связанное с обрабатываемым узел, проверяется на наличие в пользовательском вводе. Как только находится слово, присутствующее в обеих структурах, то потомки этого узла добавляются в голову дека (инициирование обхода в глубину) и увеличивается счётчик количества совпавших слов.

Так происходит до тех пор, пока дек не опустеет. На рисунке 2.17 подроб-

но изложен этот алгоритм.

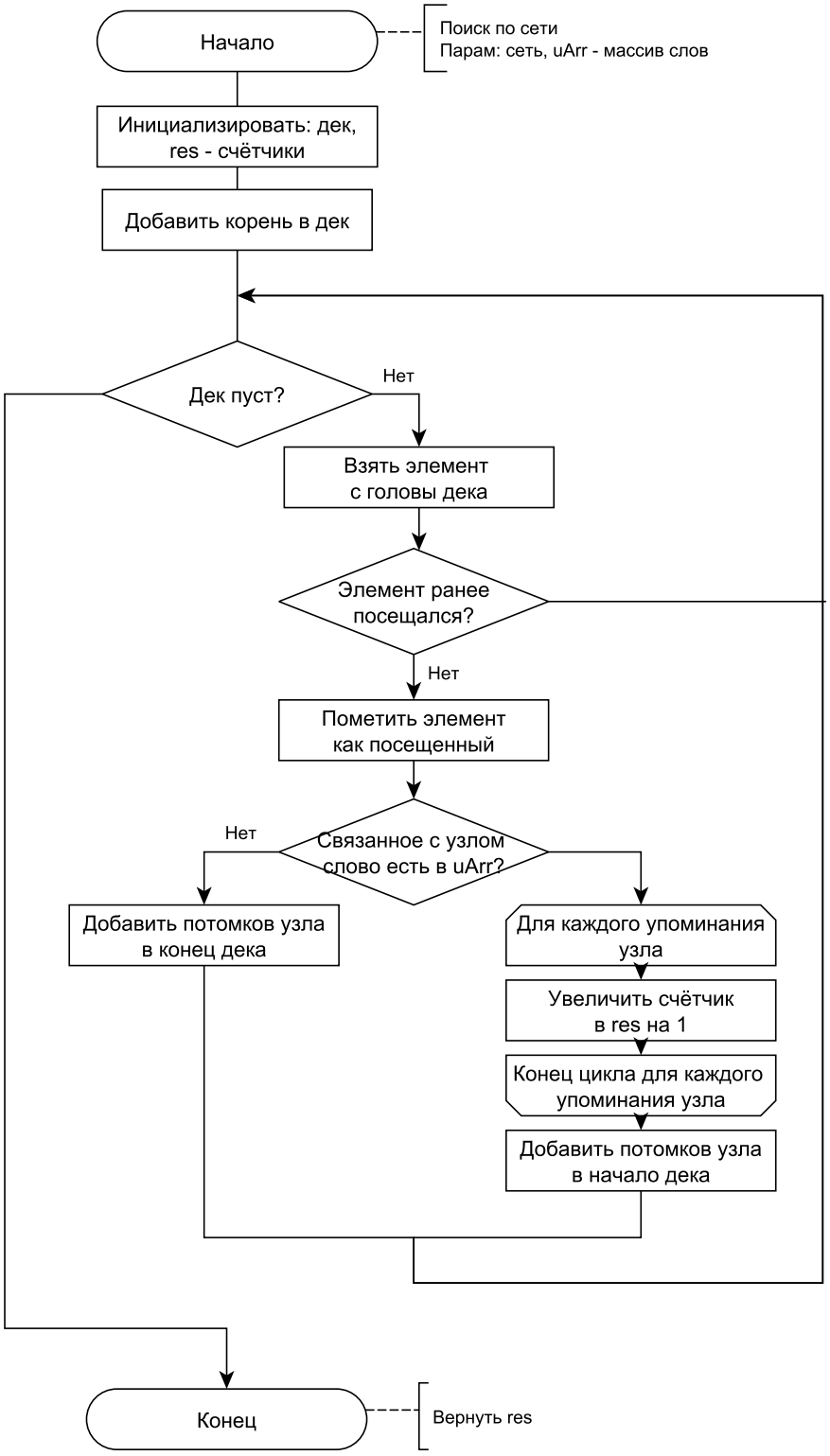


Рисунок 2.17 – Схема алгоритма поиска по сети.

2.5 ER-диаграмма

На рисунках 2.18-2.19 представлены ER-диаграммы для двух рассматриваемых онтологий.

Сущность ключевого слова (Word) содержит такие поля, как название (name) и вес (weight). Из таких элементов состоит сущность Term (термин), в нём также хранится само название термина.

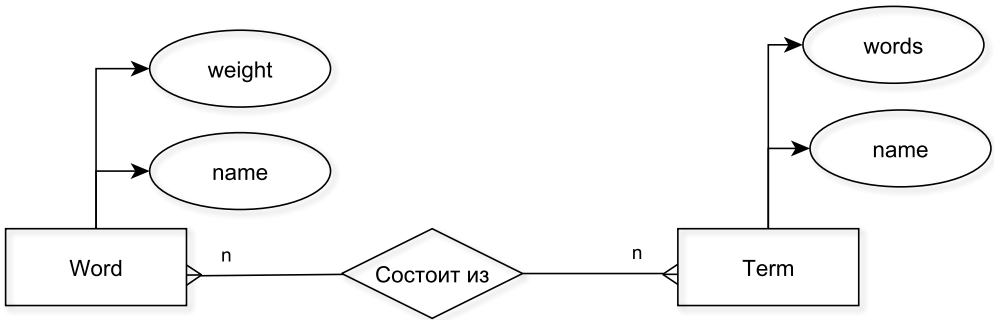


Рисунок 2.18 – ER-диаграмма сущностей статистической онтологии.

NodeTerm – узел графа, который помимо названия объекта, с которым он связан, содержит информацию о его признаках, действиях, свойствах и т.д. Кроме того, хранится информация о терминах, в которых было употреблено данное слово (mentions).

Согласно ранее установленным условиям, помимо того, что один граф может состоять из нескольких узлов, один узел может относиться к нескольким графам. Сущность сети включает как идентификационную информацию, так и информацию о её составляющих.

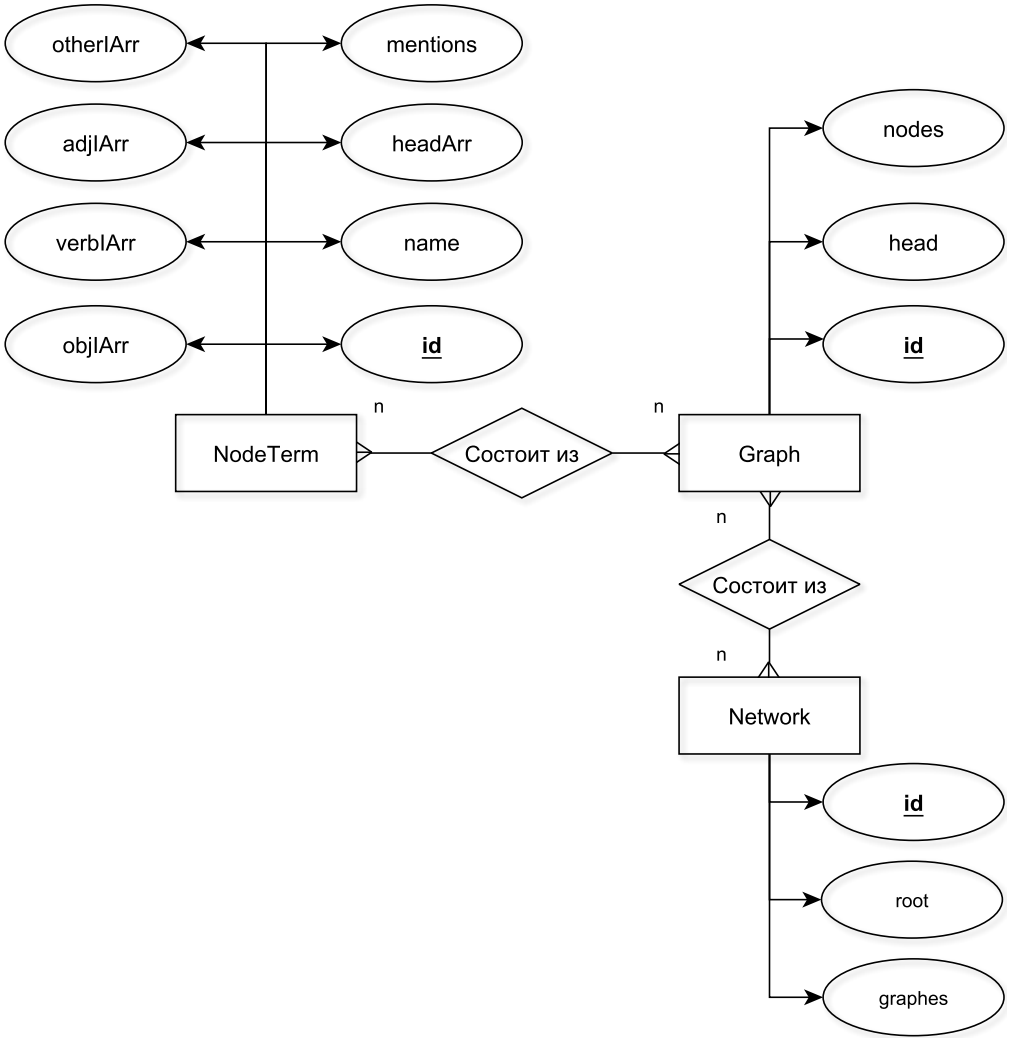


Рисунок 2.19 – ER-диаграмма сущностей онтологии на синтаксических графах.

2.6 Use-case диаграмма

На рисунке 2.20 продемонстрирована Use-case диаграмма, на которой наглядно показаны возможности каждого из участников. Выделяются две роли: пользователь и администратор.

Для обоих предлагается два способа ввести запрос: через текстовое поле или через голосовой ввод. У обоих есть возможность просмотреть подробные результаты запроса и локальную сеть запроса, если она была использована в методе.

Администратор, в отличие от пользователя, может вносить изменения в обе онтологии, и менять данные как по отдельным терминам, так и по всем

сразу.

Дополнительно он может увидеть наглядное изображение всей сети и определения терминов из словаря.

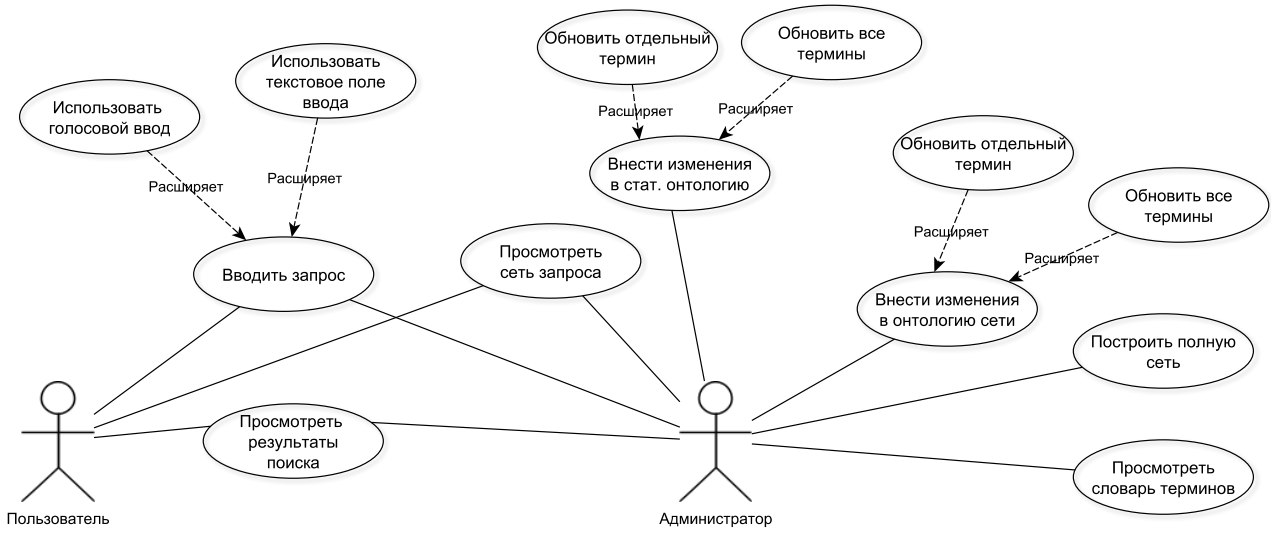


Рисунок 2.20 – Use-case диаграмма.

Выводы

В текущем разделе был определён формат входных и выходных данных, предоставлены IDEF0 схемы, подробные схемы основных алгоритмов, use-case диаграмма.

3 Технологическая часть

3.1 Выбор средств программной реализации

3.1.1 Основные средства

В качестве языка программирования был выбран Python 3 [32], ввиду нескольких причин.

- Текущая работа подразумевает активное взаимодействие с ЕЯ, в частности, с русским языком, поэтому необходимы средства, позволяющие обрабатывать соответствующие тексты. На Python написано большое количество библиотек для NLP (Natural Language Processing – обработка естественного языка).
- Также язык поддерживает объектно-ориентированный подход, что важно, поскольку в процессе реализации подразумевается использование этой методологии, позволяющей разрабатывать хорошо организованную и просто модифицируемую структуру приложения.
- Кроме того, предоставляются библиотеки для создания графического интерфейса, визуализации графов, которые планируется использовать для отладки и наглядной демонстрации работы приложения.
- В дополнение, в процессе обучения был накоплен существенный опыт в использовании этого языка программирования.

В качестве среды разработки был выбран PyCharm [33] в силу следующих факторов.

- Она бесплатна для студентов.
- Предоставляются удобные инструменты для написания, редактирования кода, а также графический отладчик.
- Помимо этого, является хорошо знакомой средой разработки, и какие-либо проблемы с взаимодействием сведены к минимуму, что позволяет сэкономить время.

3.1.2 Вспомогательные средства

Для сбора датасета использовалась кроссплатформенная система мгновенного обмена сообщениями Discord [34]. Это было сделано по следующим причинам.

- Она бесплатна для всех пользователей.
- Поскольку необходимо было опросить большое количество людей, встречаться лично было затруднительно, поэтому гораздо удобнее и проще организовать весь процесс дистанционно, что и позволяет сделать это приложение.
- Платформа очень популярна среди молодых людей, что доказывает недавнее исследование от апреля 2022 года [35]. Это упрощает поиск удобной большинству среды.
- Для этой системы возможно написание дополнительного API в виде discord-бота, который может выполнять различные задачи. Так, для ускорения процесса фиксации слов опрашиваемых было создано дополнительное приложение, которое в реальном времени переводит речь участников в текст, заносит всю информацию в файлы и по каждому человеку создаёт базу знаний.

Для создания описанного выше discord-бота с заявленными функциями использовался язык Javascript [36], поскольку в основном для написания подобных приложений используют его, к тому же бОльшая часть кода Discord написана на этом языке программирования.

3.2 Используемые библиотеки

Для разработки графического пользовательского интерфейса привлекалась библиотека PyQt5 [37]. Qt – один из самых популярных кроссплатформенных графических фреймворков [38], поэтому кроме документации, описано множество примеров его использования. Для наглядной разработки GUI при-

влекалась среда Qt Designer [39].

Для упрощения контроля над тем, правильно ли формируются графы и сети, используется библиотека NetworkX [40], позволяющая визуализировать подобные структуры.

В приложении и discord-боте для перевода речи пользователя в текст используется библиотека Speech Recognition [41]. Это инструмент от таких компаний, как Google, Microsoft, IBM и др. Для работы используется стандартный Google Speech API.

К другой библиотеке, Natasha [42], происходит обращение с целью выделения словосочетаний в предложениях. Natasha позволяет с помощью готовых правил решать базовые задачи NLP для русского языка такие, как:

- токенизация;
- сегментация;
- определение морфологических признаков;
- лемматизация/нормализация;
- выделение словосочетаний и т.д.

Кроме того, привлекается библиотека SciPy [43], которая ориентирована на работу с большим количеством данных, содержит много функций линейной алгебры, интерполяции, масштабирования данных.

3.3 Сбор данных для формирования онтологии

Для сбора различных формулировок определений терминов был проведён устный опрос среди студентов как третьего курса бакалавриата, так и некоторых аспирантов кафедры «Комбинированные двигатели и альтернативные энергоустановки» факультета «Энергомашиностроение» с последующим переводом речи в текстовый формат посредством использования заранее разработанного discord-бота.

В общей сложности было собрано около 800 определений.

3.4 UML-диаграммы

Компонент доступа к данным

Доступ к данным реализован с помощью паттерна проектирования Repository. Соответствующая UML-диаграмма представлена на рисунке 3.1.

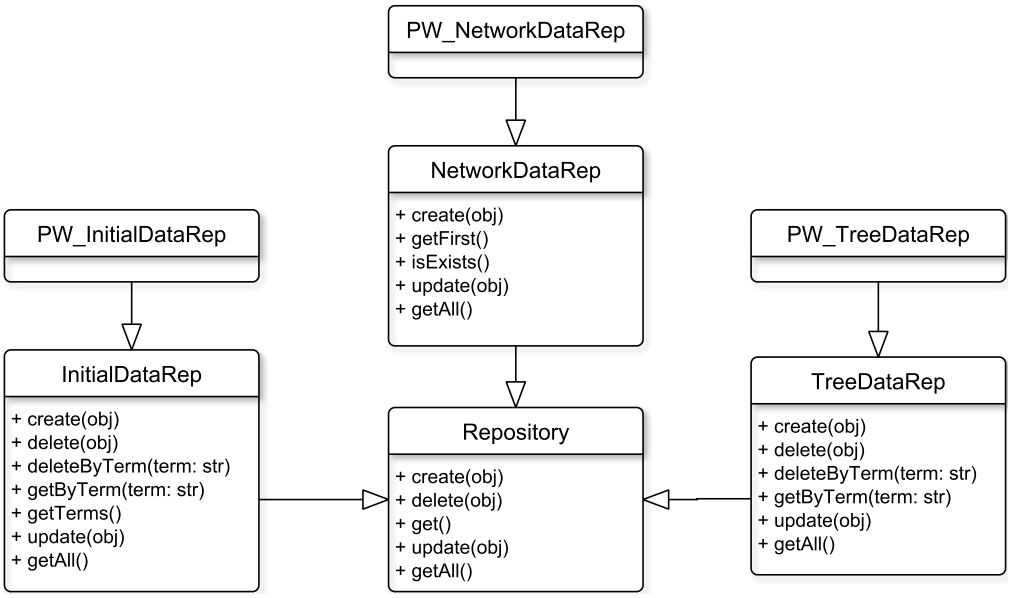


Рисунок 3.1 – UML-диаграмма компонента доступа к данным.

Компонент бизнес-логики

Этот компонент выполняет основную обработку данных, соответствующая UML-диаграмма представлена на рисунке 3.2.

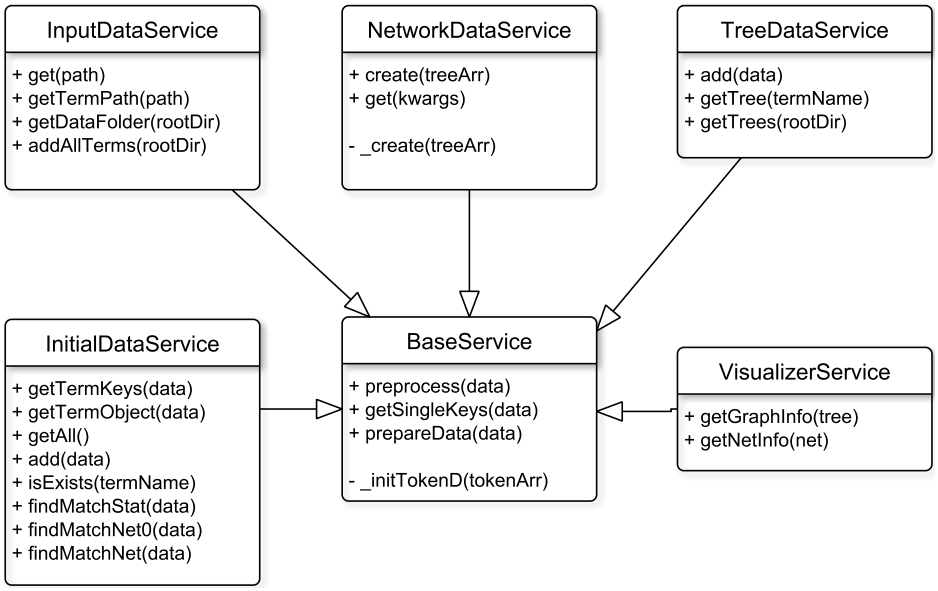


Рисунок 3.2 – UML-диаграмма компонента бизнес-логики.

Компонент представления

UML-диаграмма компонента, отвечающая за отображение, изображена на рисунке 3.3.

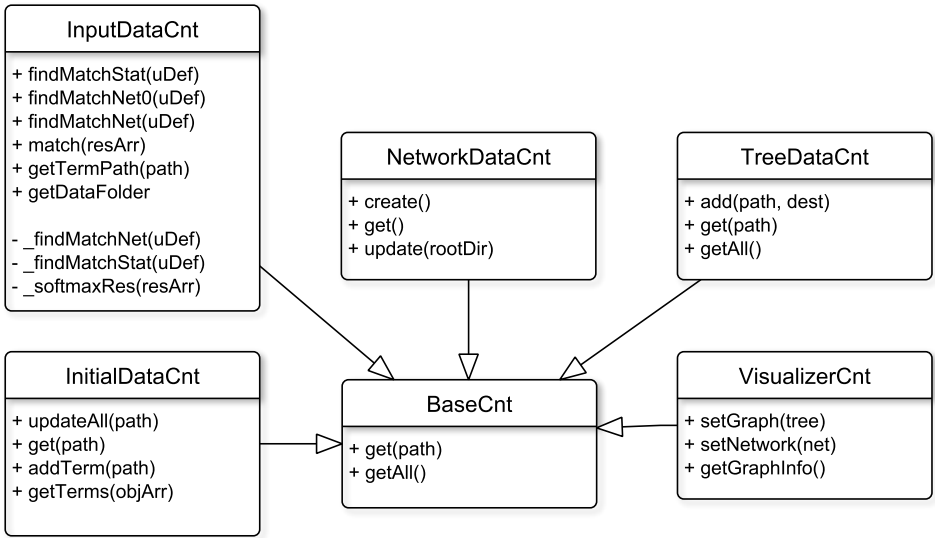


Рисунок 3.3 – UML-диаграмма компонента представления.

3.5 Интерфейс программы

На рисунках 3.4-3.6 представлен графический интерфейс пользователя.

Пользователь может ввести свой запрос в текстовое поле, либо воспользоваться голосовым вводом, нажав на кнопку микрофона. При нажатии на кноп-

ку «Готово», начнётся процесс поиска нечётких дубликатов, результат которого будет выведен в разделах «Косинусное сходство» и «Сеть синтаксических графов».

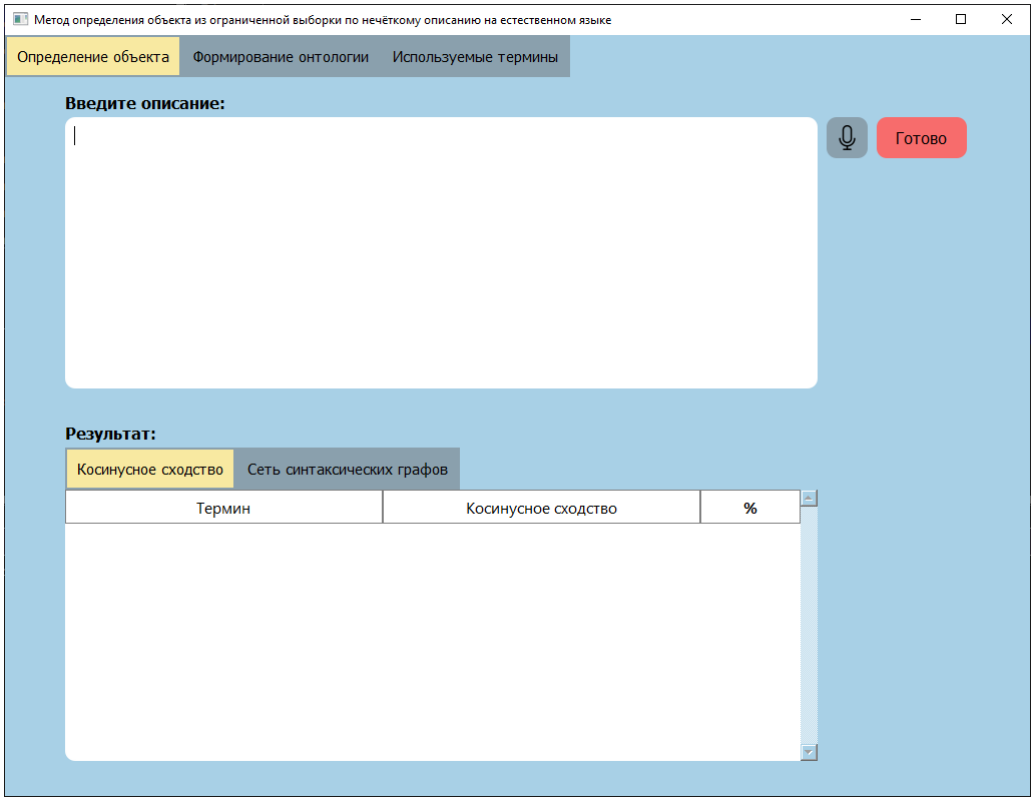


Рисунок 3.4 – Графический интерфейс (администратор/пользователь).

Последующие две страницы приложения доступны только администратору. На рисунке 3.5 изображёно окно для редактирования онтологий, помимо этого, нажав на кнопку графа в терминологическом отделе, можно увидеть как выглядит граф для выбранного термина.

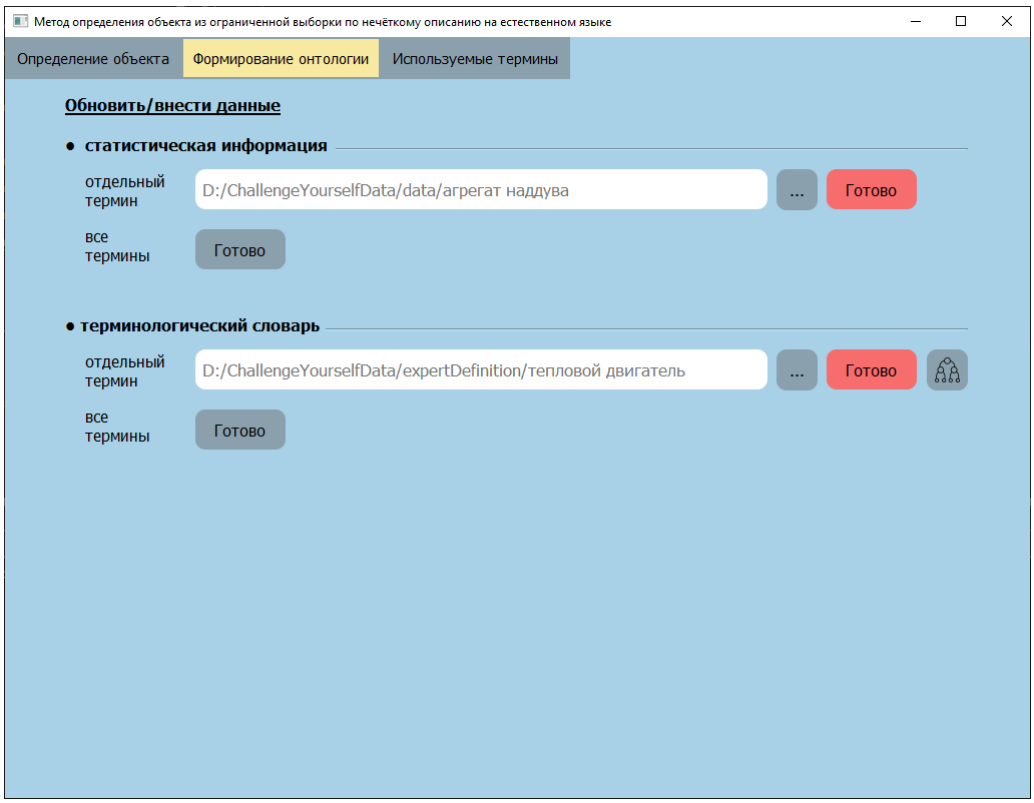


Рисунок 3.5 – Графический интерфейс (администратор).

На рисунке 3.6 представлен графический интерфейс посредством которого можно наглядно увидеть сеть для всей онтологии, а также ключевые определения терминов.

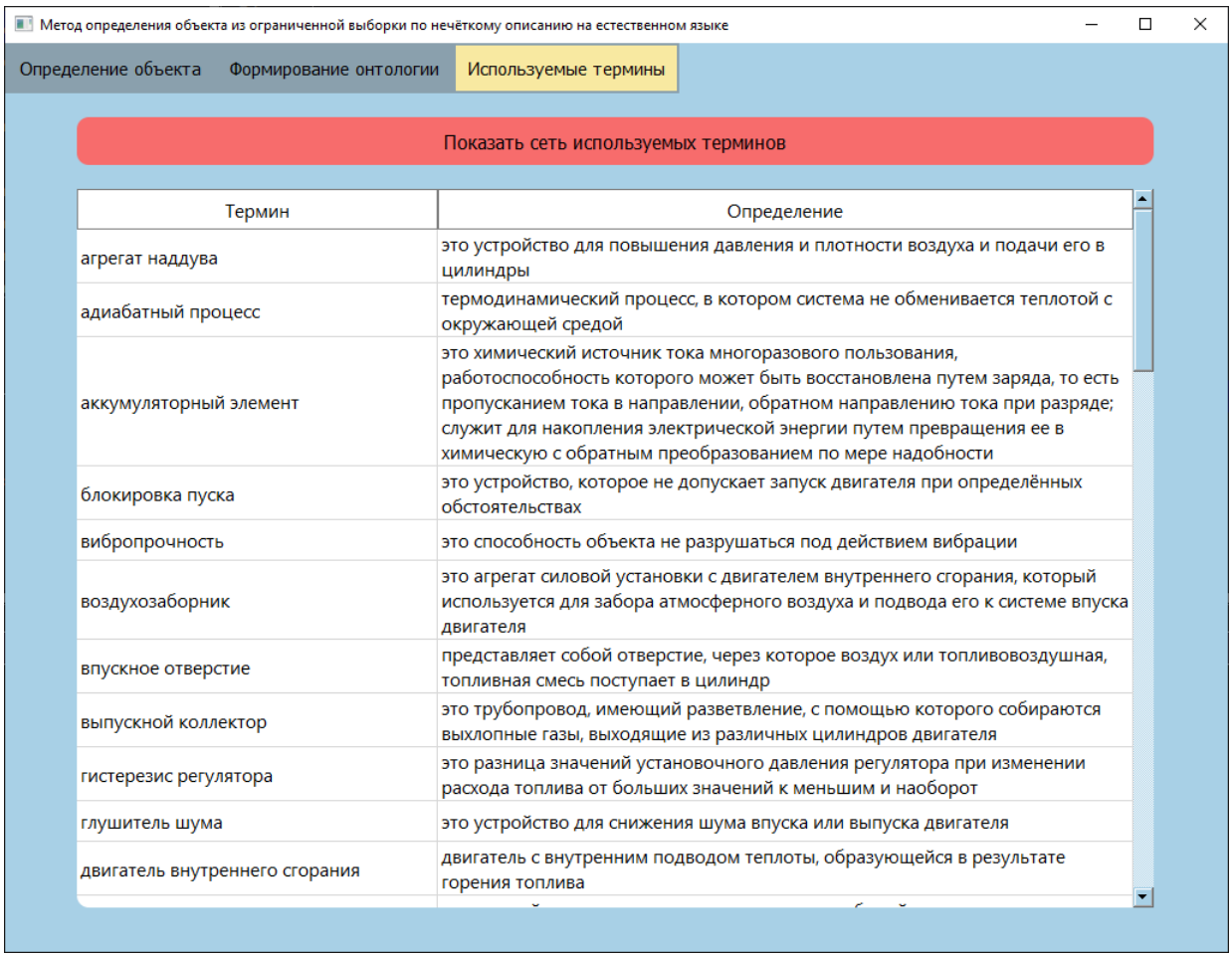


Рисунок 3.6 – Графический интерфейс (администратор).

3.6 Демонстрация работы программы

Введём с помощью голосового ввода описание понятия «сжатие»: «Как называется процесс уменьшение объёма» (рисунок 3.7).

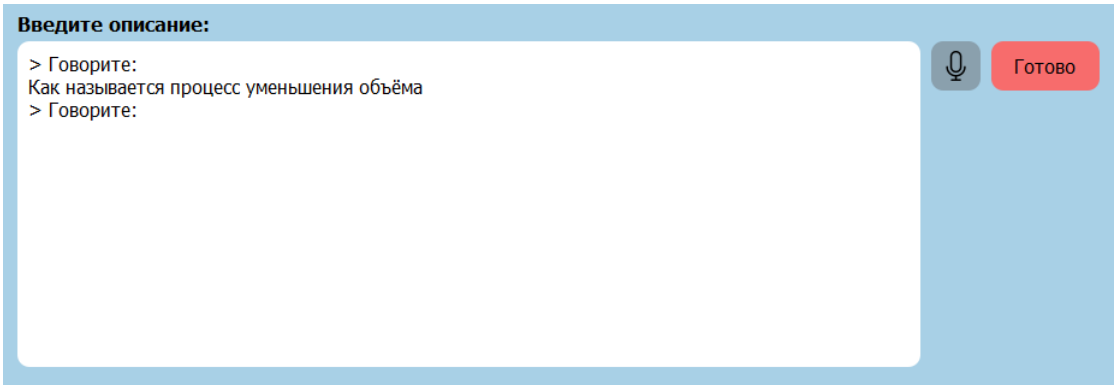


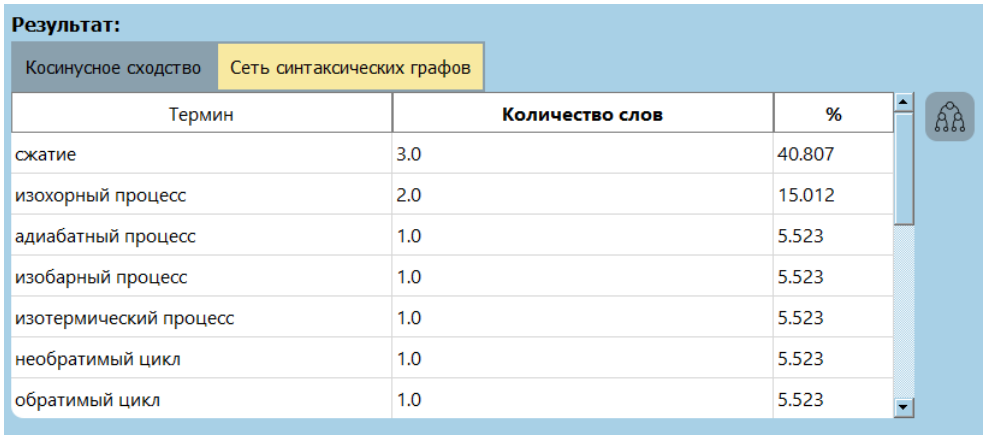
Рисунок 3.7 – Ввод данных.

Результат применения косинусного сходства представлен на рисунке 3.8. Наибольшее вычисленное расстояние – с терминами «сжатие», «изохорный процесс» (т.к. последний – термодинамический процесс, происходящий при постоянном объёме).

Термин	Косинусное сходство	%
сжатие	0.649	45.622
изохорный процесс	0.634	39.267
расширение	0.47	7.617
адиабатный процесс	0.334	1.955
изобарный процесс	0.311	1.553
изотермический процесс	0.302	1.42
рабочая камера	0.227	0.671

Рисунок 3.8 – Результат поиска с помощью косинусного сходства.

Из-за того, что самый высокий процент меньше 50, то дополнительно осуществляется поиск по семантической сети. Результат представлен на рисунке 3.9.



Термин	Количество слов	%
сжатие	3.0	40.807
изохорный процесс	2.0	15.012
адиабатный процесс	1.0	5.523
изобарный процесс	1.0	5.523
изотермический процесс	1.0	5.523
необратимый цикл	1.0	5.523
обратимый цикл	1.0	5.523

Рисунок 3.9 – Результат с помощью сети синтаксических графов.

Несмотря на то, что при таком подходе процент ниже (около 40%), в то же время у термина «изохорный процесс» он гораздо меньше, чем был ранее (только 15%). Разработанный метод верно определил описываемый объект.

Если нажать на кнопку графа, то можно увидеть сеть, которая была построена конкретно для текущего запроса пользователя (рисунок 3.10).

У администратора есть возможность посмотреть синтаксические графы для любого из загруженных терминов, для этого следует нажать на кнопку графа во второй вкладке приложения (рисунок 3.8). Так, для термина «термодинамический цикл» синтаксический граф для определения: «Термодинамический цикл – непрерывная последовательность термодинамических процессов, в результате которых рабочее тело возвращается в исходное состояние», выглядит так, как показано на рисунке 3.11.

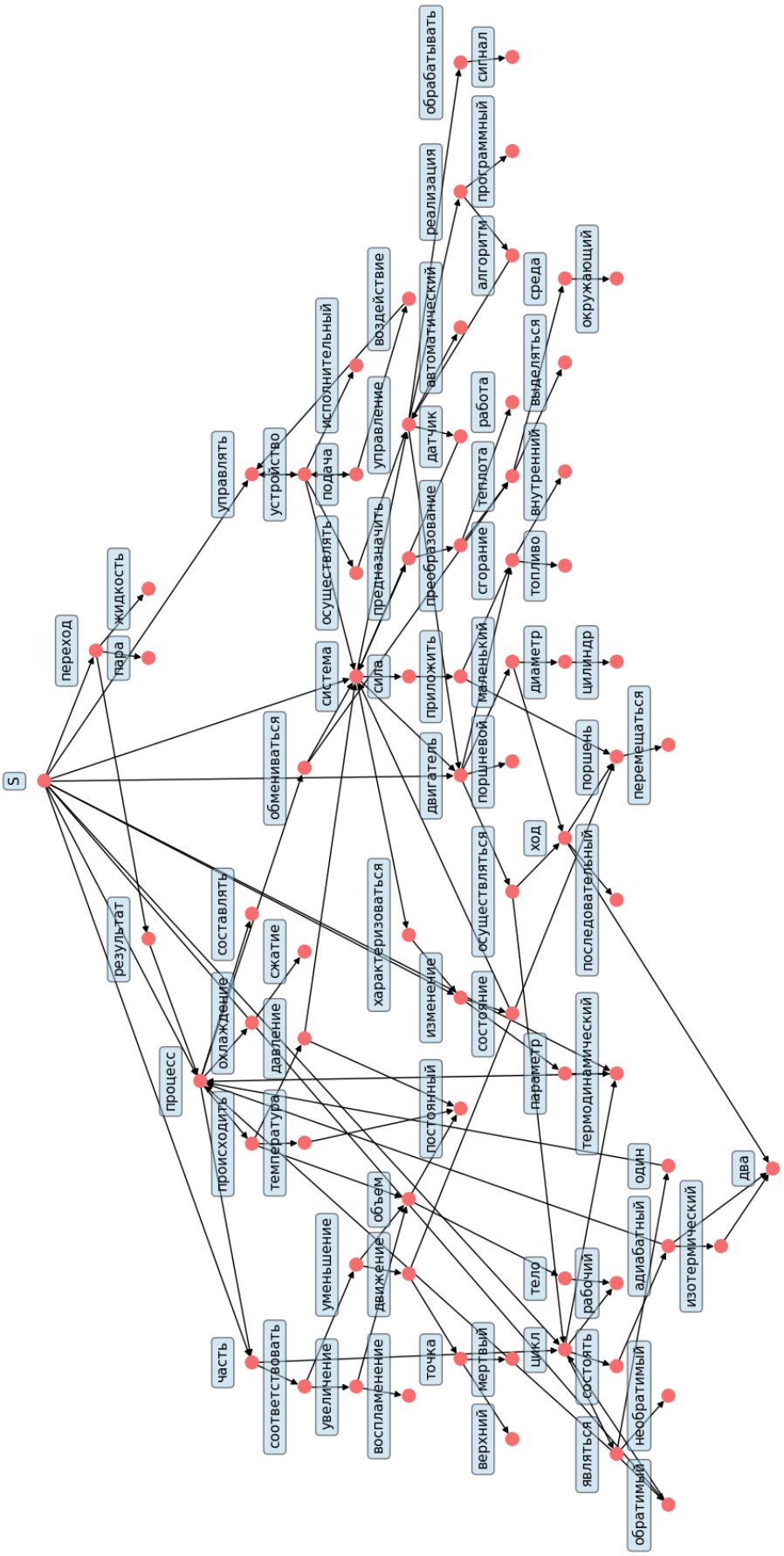


Рисунок 3.10 – Построенная для рассматриваемого запроса сеть.

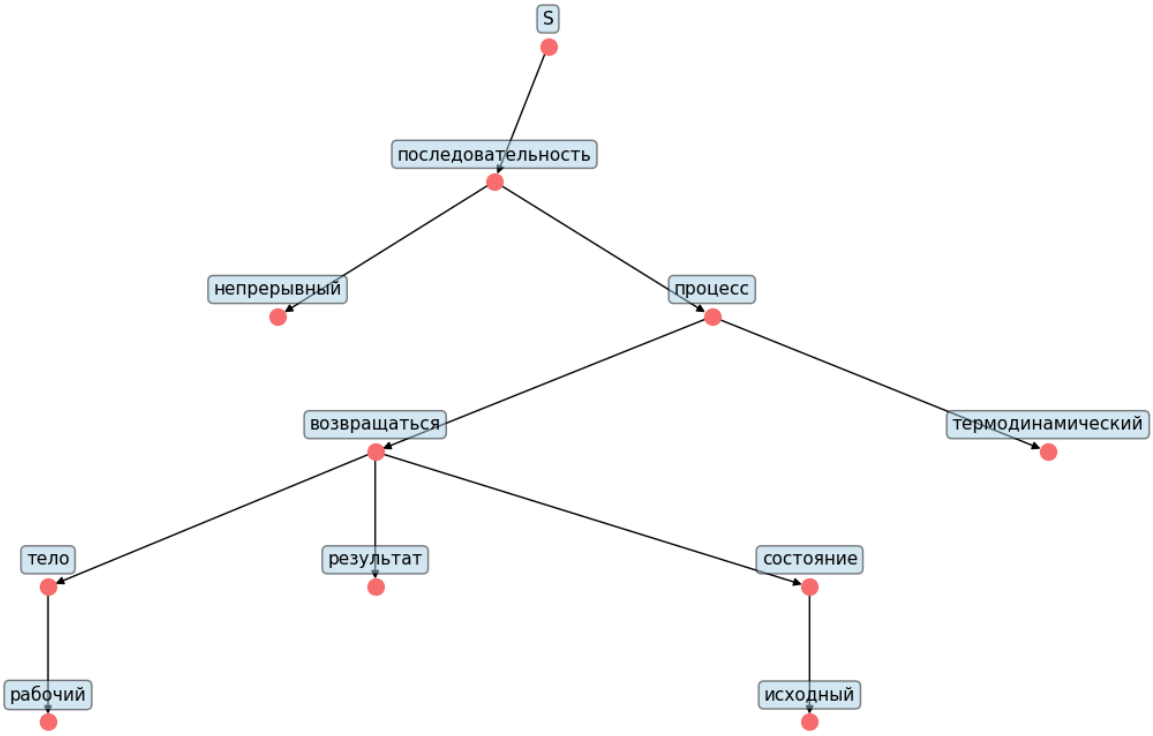


Рисунок 3.11 – Синтаксический граф для термина «термодинамический цикл».

Также есть функция просмотра сети по всей онтологии, которая есть в базе данных. Её можно увидеть, нажав на соответствующую клавишу на третьей вкладке (рисунок 3.6). Результат представлен на рисунке 3.12.

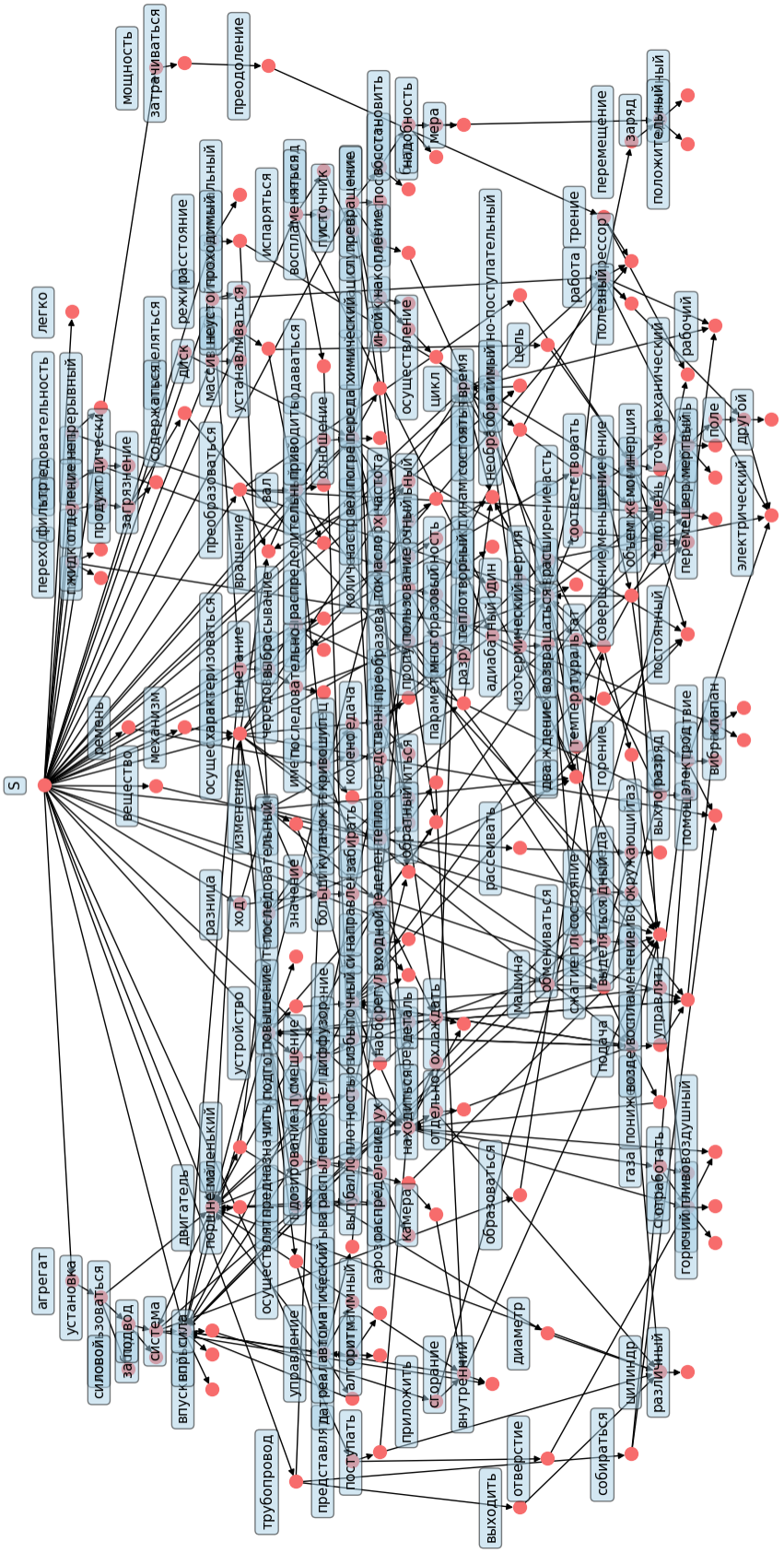


Рисунок 3.12 – Сеть по всей онтологии.

3.7 Тестирование программы

Было проведено тестирование как методом чёрного ящика (таблица 2), так и модульное (таблица 3).

Таблица 2 – Тестирование методом чёрного ящика

Описание	**Примеры**	**Ожидаемый результат**
Вход: определения, по которым создавалась онтология на базе статистических данных. **Ожидаемый результат**: Наибольшее косинусное сходство у корректного термина.	1. Уменьшение рабочего объема. 2. Цикл, протекающий как в одном, так и в обратном направлении, в замкнутой системе	1. Сжатие 2. Обратимый цикл
Вход: определения из словаря терминов, проверяемая сущность – сеть, созданная по всей онтологии. **Ожидаемый результат**: правильно определённый объект.	1. Процесс уменьшения объема рабочего тела посредством движения поршня к верхней мертвой точке 2. Термодинамический цикл, в котором все процессы являются обратимыми	1. Сжатие 2. Обратимый цикл
Вход: пустой ввод.	1. (пустая строка)	1. Сообщение об ошибке

Продолжение на следующей странице

Описание	**Примеры**	**Ожидаемый результат**
Ожидаемый результат: сообщение об ошибке.		

Таблица 3 – Модульное тестирование

Описание	**Примеры**	**Ожидаемый результат**
Проверяется метод предобработки текста.		
Вход: множество строк, включающих: • вводные фразы, междометия, слова из группы стоп-слов; • цифры и символы; • различное написание слов с буквами е и ё. **Ожидаемый результат**: каждая строка должна обрабатываться в соответствии с алгоритмом, описанным в разделе 2.4.2.	1. Я думаю, что это такой двигатель 2. Ой, это механизм для управления 3. В двигателе есть 2 специальные детали, которые управляют всем процессом!!!! 4. Объём уменьшается 5. Объем уменьшается	1. двигатель 2. механизм управление 3. двигатель специальный деталь управлять процесс 4. объем уменьшаться 5. объем уменьшаться

Продолжение на следующей странице

Описание	**Примеры**	**Ожидаемый результат**
Контролируется подсчёт косинусного расстояния.		

Вход: пары терминов с ключевыми словами. Посчитать косинусное сходство между:
• термином с самим собой;
• терминами, которые имеют частичное совпадение по ключевым словам;
• терминами, не имеющие ничего общего.
 	
Ожидаемый результат: корректные значения расстояний.
	1. термин_1 = {
	сгорание: 0.6,
	топливо: 0.3}
	
	2. термин_1 = {
	сгорание: 0.6,
	топливо: 0.3}
	термин_2 = {
	двигатель: 0.3,
	сгорание: 0.5,
	внутренний: 0.1}
	
	3. термин_1 = {
	сгорание: 0.2}
	термин_2 = {
	двигатель: 0.6}
		1. 1
		
		2. ≈ 0.75
		
		3. 0

Посчитать вес по методу TF-IDF.		

Продолжение на следующей странице

Описание	**Примеры**	**Ожидаемый результат**
Вход: несколько предложений. Для каждого слова посчитать вес для сравнения с верным ответом. Предусмотреть следующие ситуации: • есть слово, которое встречается в каждом предложении; • приведён термин, повторяющийся по несколько раз в пределах одного предложения; • наличие слова, употреблённого ровно один раз. **Ожидаемый результат**: корректные значения весов.	Последовательность состоит из шагов. На каждом шаге требуется термометр, термометр входит в состав стенда.	последовательность: 0.69 состоять: 0.69 шаг: 0.69 требоваться: 0.69 термометр: 1.38 входить: 0.69 состав: 0.69 стенд: 0.69

Все перечисленные тесты были успешно пройдены.

Выводы

В данном разделе для реализации разрабатываемого метода выбран Python в качестве основного языка программирования, Javascript – как вспомо-

гательный инструмент для реализации сопутствующего приложения для сбора датасета. В качестве среды разработки был выбран PyCharm.

Определены основные используемые библиотеки. Также изложен способ получения данных для статистического метода.

Кроме того, приведён и подробно описан интерфейс программы и продемонстрирована её работа. А также изложены основные пункты, по которым производилось тестирование ПО.

4 Исследовательская часть

4.1 Постановка задачи на исследование

Для детального изучения поведения метода при разных условиях проводится несколько исследований.

Необходимо рассмотреть, как размер выборки влияет на меру сходства.

Следует также проанализировать, как влияет размер выборки на качество определения объекта по косинусному сходству: какова доля обращений к вспомогательному методу, основанному на сети синтаксических графов, а также какой процент этих обращений приходится на неверное предположение.

При обращении к дополняющему методу необходимая сеть создаётся «на лету» на основе терминов, которые были отобраны на предыдущем шаге с применением косинусного расстояния. Следует оценить, как отличается время выполнения при таком подходе по сравнению с использованием заранее сформированной сети по всей онтологии.

Характеристики компьютера

Все проведённые эксперименты проводились на персональном компьютере с характеристиками:

- операционная система — Windows 10, 64-разрядная;
- процессор — Intel(R) Core(TM) i7-10510U CPU @ 1.80GHz 2.30 GHz;
- оперативная память — 16,00 ГБ.

4.2 Проведение исследований

4.2.1 Влияние размера выборки на меру сходства

Для проведения эксперимента использовались онтологии следующих размеров: 100, 250, 450, 600. В качестве входных данных выступают определения из предметного словаря. Исследование построено на предположении, что малый размер датасета не позволяет в достаточной степени определить сход-

ство двух определений.

Результаты исследования были сведены в таблицу 4.

Таблица 4 – Влияние размера выборки на меру сходства

Термины	**Размер выборки**				
	100	**250**	**300**	**450**	**600**
аккумуляторный элемент	0.066	0.092	0.149	0.267	0.358
блокировка пуска	0.135	0.099	0.138	0.384	0.566
гистерезис регулятора	0.113	0.105	0.110	0.121	0.152
глушитель шума	0.198	0.566	0.661	0.615	0.686
двигатель внутреннего сгорания	0.107	0.051	0.191	0.227	0.268
дефлектор	0.316	0.447	0.519	0.504	0.535
зубчатый ремень	0.0	0.159	0.194	0.286	0.347
изотермический процесс	0.408	0.615	0.815	0.97	0.991
изохорный процесс	0.671	0.811	0.949	0.975	0.984
рабочее тело	0.0	0.0	0.221	0.282	0.33

Исходя из приведённой выше таблицы можно сделать следующие выводы.

- Как правило, с увеличением размера выборки увеличивается и мера сходства между определением из онтологии и соответствующим запросом, имеет место накопительный характер формирования базы знаний, что можно увидеть на примере терминов «аккумуляторный элемент», «глушитель шума».
- Случаи, когда накопление данных приводит к обратному результату: уменьшение меры, объясняются человеческим фактором. Каждый из

опрашиваемых давал определение в той формулировке, которая казалась ему наиболее понятной и правильной, таким образом, термины описывались с разных сторон. Примером может послужить «блокировка пуска».

- Также в онтологии есть термины, которые вызвали большое затруднение у опрашиваемых, среди них «гистерезис регулятора». Многие студенты не знали, что это такое, либо ошибались в своих предположениях, поэтому косинусная мера сходства с увеличением выборки росла незначительно.
- Кроме того, существуют многозначные, сложные термины (например, «двигатель внутреннего сгорания»). В силу этого выявить схожие формулировки, ключевые слова довольно сложно. Определение из онтологии таких терминов, как правило, содержит много слов с малым весом, что определяет малую меру сходства с запросом пользователя.
- При вычислении косинусного сходства между определением из онтологии и запросом пользователя возможен случай, когда мера сходства равна 0. Это означает, что у рассматриваемой пары определений нет ничего общего. Такая ситуация возможна в двух случаях: когда сравниваются определения разных, непохожих терминов и когда онтология не достаточно полная и качественная. Похожую ситуация можно наблюдать с терминами «зубчатый ремень» и «рабочее тело».
- Существует ряд терминов, формулировки определений которых однозначны, известны большинству. В таком случае, мера сходства даже при маленьком объёме онтологии будет принимать существенно бóльшее значение, нежели остальные термины, которые были описаны выше. А с увеличением выборки всё больше будет расти. В качестве примера можно привести «изотермический процесс» и «изохорный процесс».

4.2.2 Влияние размера выборки на процент обращения к вспомогательному методу

Исследование проводилось на датасетах размерами 100, 150, 250, 400 и 650 определений.

Для всех наборов данных вычислялись ключевые слова и их вес в рамках рассматриваемой онтологии для каждого из 50 терминов. Затем проверялись определения из предметного словаря на каждой из созданной онтологии.

Каждый раз, когда результат применения статистического метода оценки не проходит по критерию принятия решения, фиксируется обращение к дополняющему методу. Для получения статистики по неверным результатам, полученный ответ сравнивается с правильным.

В итоге, по полученным значениям строится диаграмма, представленная на рисунке 4.1.

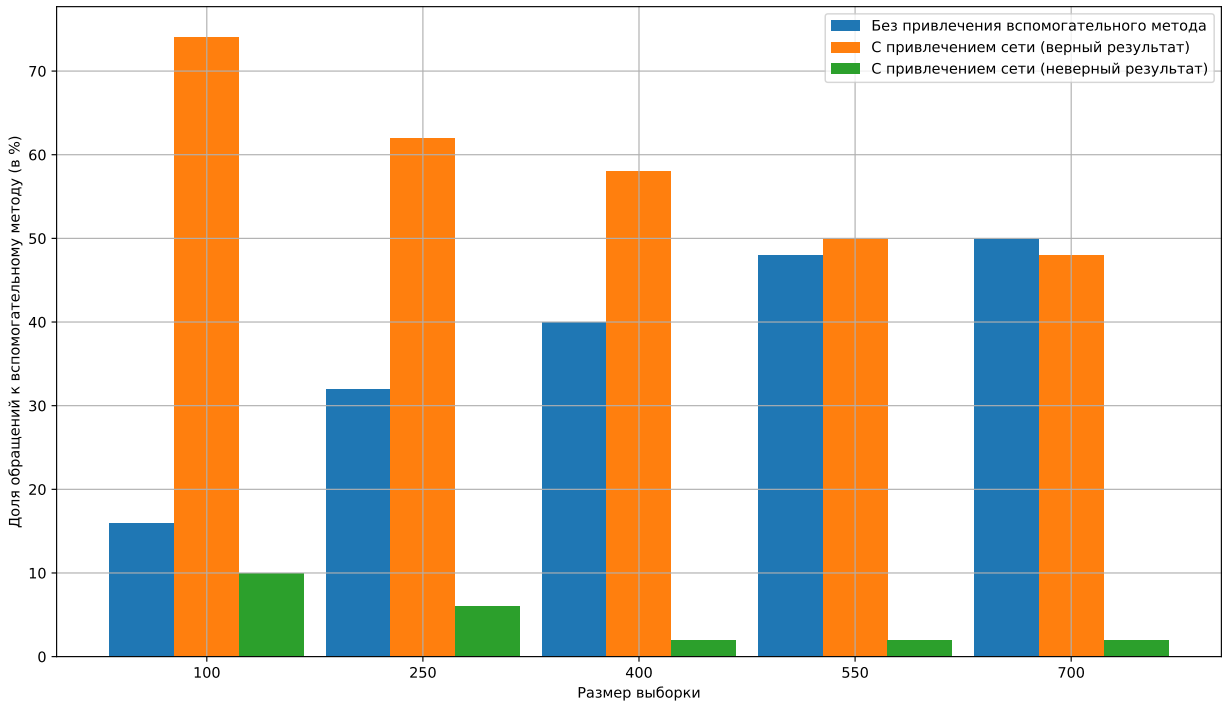


Рисунок 4.1 – Влияние размера выборки на процент обращения к вспомогательному методу.

По диаграмме можно сделать следующие выводы.

- На маленьком датасете (100 определений) доля обращений к дополняющему методу очень высока (более 80%), из них 10% приходится на неверный результат.
- В целом, с увеличением размера данных корректность работы статистического метода увеличивается, о чём свидетельствует увеличивающийся процент. Так, доля запросов, которые обрабатываются без привлечения вспомогательного метода при объёме данных в 700 определений больше примерно на 34%, чем при 100 элементах, что потенциально сокращает время обработки запроса пользователя.
- Аналогичная ситуация наблюдается с процентом неправильных ответов, которые даются при обработке вспомогательным методом, он также снижается. Это обуславливается, прежде всего, тем, что алгоритм косинусного сходства с увеличением выборки лучше идентифицирует потенциально верные термины, из которых на втором этапе алгоритма строится сеть.

4.2.3 Сравнение времени поиска по частичной и полной сети

В качестве входных данных также используются определения из предметного словаря, и только те, где результат применения косинусного сходства не удовлетворяет критерию принятия решений.

Исследование проводится на онтологии максимального размера (около 800 определений).

Для сопоставления времени выполнения используется библиотека time [44].

Количество узлов в сети, которая создаётся «на лету», определяется результатом статистического метода. Полная сеть включает в себя все термины онтологии, формируется заранее и только один раз, а затем хранится в сопутствующей базе данных, откуда извлекается для проведения этого исследования. В отличие от подхода с использованием полной сети, в частичной сети помимо поиска ещё необходимо учитывать время на формирование этой сети.

В результате была получена диаграмма, изображённая на рисунке 4.2.

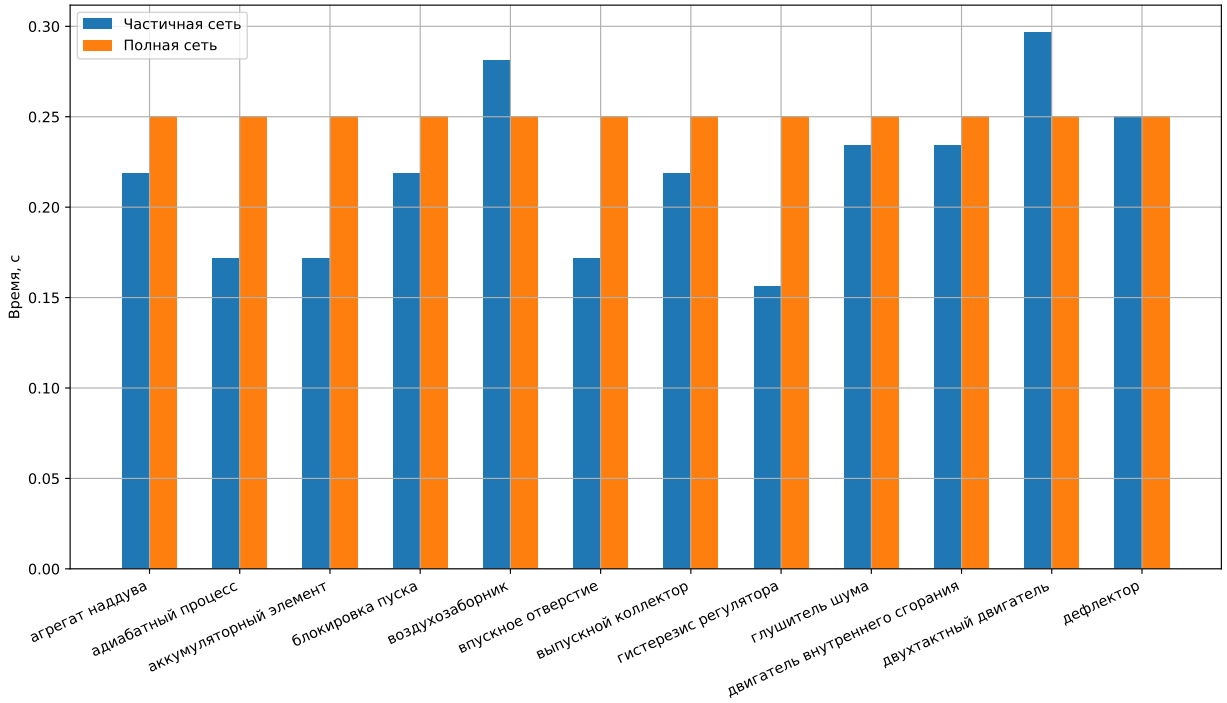


Рисунок 4.2 – Сравнение времени поиска по частичной и полной сети.

По диаграмме можно сделать следующие выводы.

- Как правило, подход с использованием частичной сети затрачивает меньше времени, несмотря на то, что дополнительно осуществляется формирование сети.
- В силу того, что полная сеть постоянна от запроса к запросу, то время обработки ожидаемо одинаковое на всех терминах.
- Несмотря на то, что в большинстве случаев неполная сеть быстрее обрабатывает запрос, иногда наблюдается обратная динамика, например, это касается терминов «воздухозаборник», «двухтактный двигатель». Объяснятся тем, что помимо поиска учитывается формирование сети. Соответственно, чем больше потенциально верных терминов было отобрано на этапе определений косинусных сходств, тем больше времени потребуется сети для ответа на запрос.

Выводы

Было проведено несколько исследований на предмет изучения поведения разработанного метода при разных условиях. Было проведено несколько испытаний на предмет того:

- как размер выборки влияет в общем на меру сходства между определением из онтологии и запросом пользователя;
- как размер выборки оказывает влияние на процент обращения к вспомогательному методу;
- стоит ли использовать частичную сеть, которая строится по ходу выполнения программы, когда можно использовать заранее подготовленную по всей онтологии сеть.

В результате всех указанных исследований можно сделать общий вывод.

- Если привлекать качественную выборку (т.е. включающую в основном правильные, полные определения), то с увеличением выборки, растёт и мера сходства.
- Также с увеличением выборки, уменьшается доля обращений к вспомогательному методу на основе сети, что позволяет сократить время обработки запроса. Кроме того, сокращается и число случаев, когда сеть определяет термин неверно.
- Несмотря на то, что тратится дополнительно время на формирование частичной сети, такой подход всё равно обрабатывает запрос быстрее, чем это делается в полной сети, построенной заранее.

ЗАКЛЮЧЕНИЕ

Таким образом, в рамках текущей выпускной квалификационной работы был разработан и реализован метод определения объекта из ограниченной выборки по нечёткому описанию на естественном языке. Объём проделанной работы соответствует требованиям технического задания.

Разработанное приложение позволяет:

- определять объект из ограниченной выборки по нечёткому описанию на естественном языке, запрос может быть изложен как через текстовое поле, так и через голосовой ввод;
- обновлять из интерфейса онтологию, построенную как на статистических данных, так и на графовых структурах;
- наглядно демонстрировать строение синтаксических графов, как в составе сети, так и по отдельности.

В результате проделанной работы были выполнены все поставленные задачи.

- Была проанализирована предметная область, проведён сравнительный анализ существующих методов решения, выявлены основные преимущества и недостатки.
- Также, рассмотрены особенности работы с текстами на естественном языке, обоснована необходимость в предварительной обработке.
- Описан принцип формирования онтологии и основные ограничения для определения ключевых слов.
- В результате проведённого предварительного анализа, определены основные этапы поиска нечётких дубликатов, а также критерий принятия решения об использовании вспомогательного метода.
- Формализованы входные и выходные данные метода.
- Пошагово описана структура реализуемого алгоритма.
- Разработано и протестировано программное обеспечение, демонстрирующее работу данного метода.

- Проведено исследование поведения алгоритма при различных входных данных и онтологиях.

В качестве направлений дальнейшей работы можно выделить следующие:

- дальнейшее увеличение датасета;
- использование параллельных вычислений при нахождении косинусного сходства запроса и составляющих онтологии;
- определение критерия досрочного выхода из процедуры поиска в сети синтаксических графов в целях уменьшения времени обработки запроса.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. IDC [Электронный ресурс]. – Режим доступа: https://www.idc.com/ (Дата обращения: 12.11.2021).

2. Data Creation and Replication Will Grow at a Faster Rate than Installed Storage Capacity, According to the IDC Global DataSphere and StorageSphere Forecasts [Электронный ресурс]. – Режим доступа: https://www.idc.com/getdoc.jsp?containerId=prUS47560321 (Дата обращения: 12.11.2021).

3. Data Age 2025: the datasphere and data-readiness from edge to core [Электронный ресурс]. – Режим доступа: https://www.i-scoop.eu/big-data-action-value-context/data-age-2025-datasphere/ (Дата обращения: 12.11.2021).

4. How the pandemic impacted data creation and storage [Электронный ресурс]. – Режим доступа: https://www.i-scoop.eu/big-data-action-value-context/data-storage-creation/ (Дата обращения: 12.11.2021).

5. Еникеев, Р. Д. Двигатели внутреннего сгорания. Основные термины и русско-английские соответствия : учеб. пособие для вузов / Р. Д. Еникеев, Б. П. Рудой – М. : Машиностроение, 2004. – 383 с. – Библиогр.: с. 378-382. – ISBN 5-217-03267-7.

6. Гаврилова, Т.А. Базы знаний интеллектуальных систем [Текст] / Т.А. Гаврилова, В.Ф. Хорошевский. – СПб: Питер, 2000. – 384 с. – ISBN 5-272-00071-4.

7. Autonomy. Autonomy Technology Whitepaper. – 1998. [Электронный ресурс]. – Режим доступа: http://www.autonomy.com (Дата обращения: 26.11.2021).

8. Олейник, Е. HP Autonomy IDOL: анализ совсем неструктурированных данных / Storage News – 2012. – № 3 (51). – С. 28-31 / [Электронный ресурс]. –

Режим доступа: http://www.storagenews.ru/51/HP_Autonomy_51.pdf (Дата обращения: 20.11.2021).

9. Громов, Ю.Ю. Интеллектуальные информационные системы и технологии: учебное пособие [Текст] / Ю.Ю. Громов, О.Г. Иванова, В.В. Алексеев и др. – Тамбов: Изд-во ФГБОУ ВПО «ТГТУ», 2013. – 244 с. – ISBN 978-5-8265-1178-7.

10. Макеева, Л. Б. Язык, онтология и реализм. [Текст] / Л. Б. Макеева; Нац. исслед. ун-т «Высшая школа экономики». – М.: Изд. дом Высшей школы экономики, 2011. – 310, [2] с. – 600 экз. – ISBN 978-5-7598-0802-2 (в пер.).

11. Gruber, T. R. A Translation Approach to Portable Ontologies. – Knowledge Acquisition – 1993. – № 5(2). – p. 199–220.

12. Noy, N.F. Ontology Development 101: A Guide to Creating Your First Ontology / N.F. Noy, D.L. McGuinness – Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report. – 2001. – SMI-2001-0880, p. 1-25.

13. Пальчунов, Д.Е. Решение задачи поиска информации на основе онтологий [Текст] / Бизнес-информатика – 2008. – № 1. – С. 3-13.

14. Лингвистический энциклопедический словарь [Текст] / Под ред. В. Н. Ярцевой. – М.: Советская энциклопедия, 1990. – 685 с.

15. Большакова, Е.И Автоматическая обработка текстов на естественном языке и анализ данных : учеб. пособие [Текст] / Е.И. Большакова, К.В. Воронцов, Н.Э. Ефремова, Э.С. Клышинский, Н.В. Лукашевич, А.С. Сапин – М.: Изд-во НИУ ВШЭ, 2017. – 269 с. – ISBN 978-5-9909752-1-7.

16. Elizabeth, D. Natural Language Processing. – Center for Natural Language Processing. – 2001.

17. Srividhya, V. Evaluating Preprocessing Techniques in Text Categorization / V. Srividhya, R. Anitha – International Journal of Computer Science and Application Issue – 2010 – p. 49-51. – ISSN 0974-0767.

18. Большакова, Е.И. Автоматическая обработка текстов на естественном языке и компьютерная лингвистика : учеб. пособие [Текст] / Е.И. Большакова, Э.С. Клышинский, Д.В. Ландэ, А.А. Носков, О.В. Пескова, Е.В. Ягунова —- М.: МИЭМ, 2011. —- 272 с. – ISBN 978–5–94506–294–8.

19. Пархоменко, П. А. Обзор и экспериментальное сравнение методов кластеризации текстов [Текст]/ П. А. Пархоменко, А. А. Григорьев, Н. А. Астраханцев – Труды ИСП РАН, 2017 – том 29, выпуск 2 – С. 161–200.

20. Akiko Aizawa An information-theoretic perspective of tf–idf measures – Information Processing and Management. – 2003 – p. 45-65.

21. Ramos, J. Using TF-IDF to Determine Word Relevance in Document Queries [Электронный ресурс]. – Режим доступа: https://www.researchgate.net/publication/228818851_Using_TF_IDF_to_determine_word_relevance_in_document_queries (Дата обращения: 02.12.2021).

22. Зиберт, А.О. Разработка системы определения наличия заимствований в работах студентов высших учебных заведений. Алгоритмы поиска нечетких дубликатов [Текст] / А.О. Зиберт, В.И. Хрусталев – Universum: Технические науки : электрон. научн. журн. – 2014. – № 3 (4).

23. Квашина, Ю.А. Методы поиска дубликатов скомпонованных текстов научной стилистики [Текст] / Ю.А. Квашина. – Технологический аудит. – 2013. – № 3/1(11) – С. 16-20.

24. Зеленков, Ю.Г. Сравнительный анализ методов определения нечетких дубликатов для WEB-документов [Текст] / Ю.Г. Зеленков, И.В. Сегалович – Труды 9-ой Всероссийской научной конференции «Электронные библиотеки:

перспективные методы и технологии, электронные коллекции» RCDL’2007: Сб. работ участников конкурса. – Т. 1. – Переславль Залесский: «Университет города Переславля», 2007. – С. 166—174.

25. Цимбалов, А.В. Метод шинглов [Текст] / А.В. Цимбалов, О.В. Золотарев – Вестник. – 2016. – Серия «Сложные системы: модели, анализ и управление». Выпуск 4. – С. 72-79.

26. Преображенский, Ю.П. О методах создания рекомендательных систем [Текст] / Ю.П. Преображенский, В. М. Коновалов – Вестник Воронежского института высоких технологий. – 2019. – № 4(31). – С.75-79.

27. Бабкин, Э.А. Принципы и алгоритмы искусственного интеллекта: Монография / Э.А. Бабкин, О.Р. Козырев, И.В. Куркина. – Н. Новгород: Нижегород. гос. техн. ун-т. 2006. 132 с.

28. Теньер Л. Основы структурного синтаксиса: Пер. с фр. – Прогресс, 1988.

29. Национальный корпус русского языка. 2003—2022 [Электронный ресурс]. – Режим доступа: ruscorpora.ru (Дата обращения: 04.04.2022).

30. Национальный корпус русского языка. Синтаксический корпус. 2003—2022 [Электронный ресурс]. – Режим доступа: https://ruscorpora.ru/new/search-syntax.html (Дата обращения: 04.04.2022).

31. Маршаков Д. В. СРАВНЕНИЕ РЕЗУЛЬТАТОВ НЕЙРОСЕТЕВОЙ КЛАССИФИКАЦИИ С ПРИМЕНЕНИЕМ SOFTMAX И ФУНКЦИИ РАССТОЯНИЯ //Математические методы в технологиях и технике. – 2021. – №. 8. – С. 75-78.

32. Документация по Python 3 [Электронный ресурс]. Режим доступа: https://docs.python.org/3/ (Дата обращения 01.02.2022)

33. Документация по PyCharm [Электронный ресурс]. Режим доступа: https://www.jetbrains.com/pycharm/guide/tips/quick-docs/ (Дата обращения 01.02.2022)

34. Документация по Discord [Электронный ресурс]. Режим доступа: https://support.discord.com/hc/ru (Дата обращения 05.02.2022)

35. Аналитика трафика и доля рынка Discord [Электронный ресурс]. Режим доступа: https://www.similarweb.com/ru/website/discord.com/ (Дата обращения 20.05.2022)

36. Документация по Javascript [Электронный ресурс]. Режим доступа: https://javascript.ru/manual (Дата обращения 02.02.2022)

37. Документация по PyQt5 [Электронный ресурс]. Режим доступа: https://www.riverbankcomputing.com/static/Docs/PyQt5/ (Дата обращения 04.04.2022)

38. Документация по Qt [Электронный ресурс]. Режим доступа: https://doc.qt.io/ (Дата обращения 04.04.2022)

39. Документация по Qt Designer [Электронный ресурс]. Режим доступа: https://doc.qt.io/qt-5/qtdesigner-manual.html (Дата обращения 04.04.2022)

40. Документация по NetworkX [Электронный ресурс]. Режим доступа: https://networkx.org/documentation/stable/tutorial.html (Дата обращения 28.03.2022)

41. Документация по SpeechRecognition [Электронный ресурс]. Режим доступа: https://pypi.org/project/SpeechRecognition/ (Дата обращения 15.04.2022)

42. Документация по Natasha [Электронный ресурс]. Режим доступа: https://github.com/natasha/natasha (Дата обращения 07.04.2022)

43. Документация по SciPy [Электронный ресурс]. Режим доступа: https://docs.scipy.org/doc/scipy/ (Дата обращения 10.05.2022)

44. Документация по time [Электронный ресурс]. Режим доступа: https://docs.python.org/3/library/time.html (Дата обращения 20.05.2022)

ПРИЛОЖЕНИЕ А

Таблица 5 – Используемые термины

№	Название термина	№	Название термина
1	агрегат наддува	22	конденсация
2	адиабатный процесс	23	контроллер
3	аккумуляторный элемент	24	коэффициент полноты сгорания топлива
4	блокировка пуска	25	к-т усиления регулятора
5	вибропрочность	26	лубрикатор
6	воздухозаборник	27	маховик
7	впускное отверстие	28	механическая мощность компрессора
8	выпускной коллектор	29	необратимый цикл
9	гистерезис регулятора	30	обменник давления
10	глушитель шума	31	обратимый цикл
11	двигатель внутреннего сгорания	32	охладитель
12	двухтактный двигатель	33	помпаж компрессора
13	дефлектор	34	привод распределительного вала
14	длинноходный двигатель	35	пусковая жидкость
15	зубчатый ремень	36	рабочая камера
16	изобарный процесс	37	рабочее тело
17	изотермический процесс	38	рабочий ход
18	изохорный процесс	39	распределительный вал
19	карбюратор	40	расширение
20	коленчатый вал	41	свеча зажигания
21	компрессор	42	сжатие

Продолжение на следующей странице

№	Название термина	№	Название термина
43	термодинамический процесс	47	тепловой двигатель
44	термодинамический цикл	48	ход поршня
45	топливный фильтр	49	цикл Карно
46	форсунка	50	электрическое напряжение