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 ИУ «Информатика и системы управления»

КАФЕДРА ИУ-7 «Программное обеспечение ЭВМ и информационные технологии»

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К КУРСОВОЙ РАБОТЕ

НА ТЕМУ:

«Разработка системы извлечения многокомпонентных терминов и их переводных эквивалентов из параллельных научно-технических текстов»

Студент **ИУ7-63Б** _____ **Сапожков А. М.**

Руководитель _____ **Строганов Ю. В.**

2023 г.

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 ИУ «Информатика и системы управления»

КАФЕДРА ИУ-7 «Программное обеспечение ЭВМ и информационные технологии»

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К КУРСОВОЙ РАБОТЕ

НА ТЕМУ:

«Разработка системы извлечения многокомпонентных терминов и их переводных эквивалентов из параллельных научно-технических текстов»

Студент **ИУ7-63Б** _____ **Сапожков А. М.**

Руководитель _____ **Строганов Ю. В.**

2023 г.

Реферат

Расчетно-пояснительная записка 26 с., 3 рис., 0 табл., ? ист., ? прил.

ОПЕРАЦИОННЫЕ СИСТЕМЫ, СЕТЕВЫЕ ОПЕРАЦИОННЫЕ СИСТЕМЫ, ИНТЕРНЕТ ВЕЩЕЙ, IOT

Цель работы: изучение существующих операционных систем для устройств интернета вещей.

В данной работе проводится изучение и сравнение операционных систем для интернета вещей (IoT).

Результаты: среди рассмотренных операционных систем были выделены:

- Azure Sphere, Windows 10 IoT и Amazon FreeRTOS как наиболее функциональные и масштабируемые;
- ОСРВ МАКС и KasperskyOS как наиболее доступные с точки зрения использования прикладных служб;
- Ubuntu Core и Raspbian как наиболее адаптированные для бытового применения.

СОДЕРЖАНИЕ

Введение

Стремительное развитие и внедрение технологий искусственного интеллекта и технологий автоматической обработки текстовой информации способствуют развитию лингвистических баз данных как основы создания прикладных программных средств, проведения лингвистических исследований источников информации при решении ряда прикладных задач, где однозначная и упорядоченная терминология имеет особую значимость. Под терминологической базой данных принято понимать организованную в соответствии с определёнными правилами и поддерживаемую в памяти компьютера совокупность данных, характеризующую актуальное состояние некоторой предметной области и используемую для удовлетворения информационных потребностей пользователей [1]. Каждый термин дополнен информацией о его значении, эквивалентах в других языках, кратких формах, синонимах, сведениях об области применения. По целевому назначению терминологические базы данных разделяют на одноязычные, предназначенные для обеспечения информацией о стандартизованной и рекомендованной терминологии, и многоязычные, ориентированные на работы по переводу научно-технической литературы и документации [2] [3].

Создание терминологических баз данных представляет собой сложный и трудоемкий процесс, требующий значительного количества времени на их создание и обновление, что особенно важно для развивающихся терминологий таких предметных областей, как авиация, космонавтика, нанотехнологии, биоинженерия, информационные технологии и многих других. Одним из наиболее время-затратных процессов является ручной сбор иллюстративного материла – извлечение специальной терминологии из коллекций текстов, что требует наличия средств автоматического извлечения многокомпонентных терминов при обработке научно-технических текстов.

Существующие программные средства автоматического извлечения терминов основаны на лингвистических и статистических методах. Могут быть ис-

пользованы и методы машинного обучения [4], сложность их реализации вызвана необходимостью наличия огромных массивов обучающих данных, которые могут отсутствовать для определенной предметной области. В основе лингвистических методов лежит использование грамматики лексико-синтаксических шаблонов, представляющих собой структурные модели лингвистических конструкций [5] [6]. Статистический подход заключается в нахождении n-грамм слов по заданным частотным характеристикам [7] [8]. Гибридный подход для выделения терминологических сочетаний, объединяющий лингвистический и статистический методы, заключается в предварительном описании моделей, по которым могут быть построены термины для последующего поиска их в коллекции текстов [9].

Выравнивание терминологических единиц в параллельных текстах обычно осуществляется в два этапа: сначала выделяют терминологические единицы на каждом языке отдельно, затем один из одноязычных списков терминов-кандидатов интерпретируется как язык источника, и для каждого термина-кандидата на языке источнике предлагаются потенциально эквивалентные термины в списке терминов-кандидатов на языке перевода [10].

Целью данной работы является разработка системы извлечения многокомпонентных терминов и их переводных эквивалентов из параллельных научно-технических текстов. Для достижения поставленной цели необходимо решить следующие задачи.

1. Провести анализ предметной области и формализовать задачу.
2. Спроектировать базу данных и структуру программного обеспечения.
3. Реализовать интерфейс для доступа к базе данных.
4. Реализовать программное обеспечение, которое позволит пользователю создавать, получать и изменять сведения из разработанной базы данных.
5. Провести исследование зависимости времени выполнения запросов от использования индексов.

1 Аналитическая часть

1.1 Метод извлечения многокомпонентных терминов на основе структурных моделей

Предлагаемый метод автоматического извлечения русскоязычных многокомпонентных терминов на основе базы данных структурных моделей терминологических словосочетаний состоит из пяти основных этапов [11].

1. Анализ предложения по частям речи.
2. Удаление частей речи и их сочетаний, которые не входят в состав терминологических словосочетаний:
   — глаголы;
   — союзы;
   — местоимения;
   — частицы;
   — знаки препинания;
   — «наречие + предлог».
3. Удаление из оставшихся терминов-кандидатов на наличие стоп-слов, указанных в специальной зоне словаря, и если они есть, убираем их. Под стоп-словами понимаются слова, которые образуют широко используемые коллокации с терминами, но в совокупности не являются терминами по сути, например современная химия, рассматриваемый метод синтеза о-гликозидов.
4. Соотнесение полученных цепочек слов с шаблонами терминологических словосочетаний, которые хранятся в базе структурных моделей терминов.
5. Проверка полученных терминов-кандидатов по словарю корпуса.
   5.1. Если термин-кандидат есть в словаре, то он извлекается как термин.
   5.2. Если полученный термин-кандидат отсутствует в словаре, то он отправляется терминологу для ручной обработки.
   5.3. Если термин-кандидат состоит из нескольких слов, то производится попытка разбить его на несколько терминов.

1.2 Постановка задачи

Диаграмма, оформленная в соответствии с нотацией IDEF0 и отражающая декомпозицию алгоритма работы системы извлечения многокомпонентных терминов, представлена в приложении А. Исходя из специфики решаемой задачи, можно сформулировать требования к приложению [12].

1. Задание исходных текстов должно производиться из файла или через специально предусмотренные поля ввода.
2. Пользователю должна быть предоставлена возможность перед сохранением терминов в базу данных выполнять над ними следующие операции.
   2.1. Редактирование.
   2.2. Сопоставление переводных эквивалентов.
   2.3. Присвоение характеристик.
3. Пользователь может анализировать и редактировать добавленные в базу данных термины путём их поиска по заданным характеристикам.

1.3 Формализация ролей

Для работы с системой обязательным этапом является прохождение аутентификации. Пользователь может работать в системе под одной из следующих ролей.

1. Студент — пользователь, имеющий возможность обрабатывать тексты, сохранять выделенные термины, а также анализировать и редактировать только те термины, которые он выделил.
2. Преподаватель — пользователь, обладающий функционалом, доступным роли ”Студент а также имеющий возможность анализировать и редактировать все термины, хранящиеся в базе данных.
3. Администратор — пользователь, имеющий возможности, доступные роли ”Преподаватель а также имеющий в распоряжении специальные функции для анализа состояния системы и настройки её работы. Также администратор имеет возможность создавать аккаунты, соответствующие любой из ро-

лей в системе.

В ходе использования приложения предусмотрена возможность смены ролей путём прохождения повторной аутентификации.

Также стоит отметить, что до входа в аккаунт пользователь считается неавторизованным и не имеет доступа к функционалу системы, так как любая работа с терминами должна быть персонализирована.

На рисунках ?????-????? представлена диаграмма вариантов использования системы в соответствии с выделенными типами пользователей.

1.4 Формализация данных

С учётом выделенных структур данных и типов пользователей разрабатываемая база данных должна хранить информацию о следующих сущностях:

1.5 Анализ моделей БД

Определение БД и СУБД

1.5.1 Классификация СУБД по модели данных

Модель данных [ссылка на новую книгу Дейта] – это абстрактное, независимое, логическое определение структур данных, операторов над данными и прочего, что в совокупности составляет абстрактную систему, с которой взаимодействует пользователь.

По модели данных СУБД можно разделить на следующие типы.

1. **Дореляционные**.

   [ссылка на старую книгу Дейта] старые (дореляционные) системы можно разделить на три большие категории : системы с инвертированными списками (inverted list), иерархические (hierarchic) и сетевые (network). Примечание. Термин сетевая система в данном случае не имеет ничего общего с коммуникационной сетью, как описано в следующей главе.) В настоящей книге эти категории подробно не рассматриваются, поскольку, по крайней мере, с точки зрения технологии, их можно считать устаревшими.

1.1. **Инвертированные списки (файлы)**.

   БД на основе инвертированных списков представляет собой совокуп-

ность файлов, содержащих записи (таблиц). Для записей в файле определен некоторый порядок, диктуемый физической организацией данных. Для каждого файла может быть определено произвольное число других упорядочений на основании значений некоторых полей записей (инвертированных списков). Обычно для этого используются индексы. В такой модели данных отсутствуют ограничения целостности как таковые. Все ограничения на возможные экземпляры БД задаются теми программами, которые работают с БД. Одно из немногих ограничений, которое все-таки может присутствовать - это ограничение, задаваемое уникальным индексом.

1.2. **Иерархичекие**.

Иерархическая модель БД состоит из объектов с указателями от родительских объектов к дочерним, соединяя вместе связанную информацию. Иерархические БД могут быть представлены в виде дерева. Производительность в значительной степени зависит от подхода, выбранного самим пользователем (прикладным программистом и/или администратором базы данных).

1.3. **Сетевые**.

К основным понятиям сетевой модели БД относятся: элемент (узел), связь. Узел — это совокупность атрибутов данных, описывающих некоторый объект. Сетевые БД могут быть представлены в виде графа. В сетевой БД логика процедуры выборки данных зависит от физической организации этих данных. Поэтому эта модель не является полностью независимой от приложения. Другими словами, если необходимо изменить структуру данных, то нужно изменить и приложение.

2. **Реляционные**.

Ключевые особенности реляционной модели данных [ссылка на старую книгу Дейта]: ■ Во-первых, отметим, что определение реляционной системы требует, чтобы база данных только воспринималась пользователем как

набор таблиц. Таблицы в реляционной системе являются логическими, а не физическими структурами. На самом деле, на физическом уровне система может использовать любую из существующих структур памяти (последовательный файл, индексирование, хэширование, цепочку указателей, сжатие и т.п.), лишь бы существовала возможность отображать эти структуры в виде таблиц на логическом уровне. Данное положение можно сформулировать и по-другому: таблицы представляют собой абстракцию способа физического хранения данных, в которой все нюансы реализации на уровне физической памяти (размещение хранимых записей, упорядочение хранимых записей, кодировка хранимых данных, префиксы хранимых записей, хранимые структуры доступа, такие как индексы и т.д.) скрыты от пользователя. ■ Во-вторых, у реляционных баз данных есть одно замечательное свойство, опреде ляемое так называемым информационным принципом: все информационное наполне ние базы данных представлено одним и только одним способом, а именно — явным за данием значений, помещенных в позиции столбцов в строках таблицы. Этот метод представления — единственно возможный для реляционных баз данных (естест венно, на логическом уровне). В частности, нет никаких указателей, связывающих одну таблицу с другой.

В первом приближении реляционная модель состоит из следующих пяти компонентов [ссылка на старую книгу Дейта]. 1. Неограниченный набор скалярных типов (включая, в частности, логический тип или истинностное значение). 2. Генератор типов отношений и соответствующая интерпретация для сгенерирован ных типов отношений. 3. Возможность определения переменных отношения для указанных сгенерированных типов отношений. 4. Операция реляционного присваивания для присваивания реляционных значений указанным переменным отношения. 5. Неограниченный набор общих реляционных операторов (реляционная алгебра) для получения значений отношений из других значений отношений.

Вполне очевидно, что реляционная модель — это нечто большее, чем просто ”таблицы плюс операции сокращения, проекции и соединения хотя ее неформально довольно часто характеризуют именно таким образом.

3. **Постреляционные**.

   [Ссылка на Маркина] В связи с быстрым ростом количества данных и их усложнением возникла необходимость в поиске новых подходов к хранению и обработке, отличных от реляционных. Таким решением стала NoSQL-технология.

   Постреляционная модель является расширением реляционной модели. Она снимает ограничение неделимости данных, допуская многозначные поля, значения которых состоят из подзначений, и набор значений воспринимается как самостоятельная таблица, встроенная в главную таблицу.

1.5.2 Выбор модели БД

2 Конструкторская часть

2.1 Проектирование БД

2.1.1 Формализация сущностей системы

2.1.2 Триггеры БД?

2.1.3 Ролевая модель

2.2 Проектирование программного комплекса

2.2.1 Архитектура приложения

2.2.1.1. Монолитная архитектура

2.2.1.2. Микросервисная архитектура

2.2.1.3. Выбор архитектуры приложения

Выбираем микросервисы, потому что очень сильно выигрываем от плюсов (особенно от разных языков, но пока не говорю, каких).

2.2.2 Связь компонентов приложения

2.2.2.1. REST API

2.2.2.2. gRPC

2.2.2.3. Выбор взаимодействия компонентов приложения

gRPC

2.2.3 Структура программного комплекса

На рисунке представлена структура программного комплекса, оформленная в виде диаграммы развёртывания. Она отражает компоненты, необходимые для работы системы, а также способы их взаимодействия.

2.2.4 Паттерны проектирования

[Ссылка на книгу по паттернам]

Паттерны появились потому, что многие разработчики искали пути повышения гибкости и степени повторного использования своих программ. Найденные решения воплощены в краткой и легко применимой на практике форме. «Банда Четырех» объясняет каждый паттерн на простом примере четким и

понятным языком. Использование паттернов при разработке программных систем позволяет проектировщику перейти на более высокий уровень разработки проекта. Теперь архитектор и программист могут оперировать образными названиями паттернов и общаться на одном языке.

Паттерны проектирования упрощают повторное использование удачных проектных и архитектурных решений. Представление прошедших проверку временем методик в виде паттернов проектирования делает их более доступными для разработчиков новых систем. Паттерны проектирования помогают выбрать альтернативные решения, упрощающие повторное использование системы, и избежать тех альтернатив, которые его затрудняют. Паттерны улучшают качество документации и сопровождения существующих систем, поскольку они позволяют явно описать взаимодействия классов и объектов, а также причины, по которым система была построена так, а не иначе. Проще говоря, паттерны проектирования дают разработчику возможность быстрее найти правильный путь.

Далее будут описаны паттерны проектирования, которые будут использованы при разработке программного комплекса.

2.2.4.1. Active Record

2.2.4.2. Dependency injection

2.2.4.3. Template method

[Ссылка на книгу по паттернам]

Template Method (шаблонный метод) — паттерн поведения классов.

Шаблонный метод определяет скелет алгоритма, перекладывая ответственность за некоторые его шаги на подклассы. Это позволяет подклассам переопределять отдельные шаги алгоритма, не меняя его общей структуры.

Основные условия для применения паттерна шаблонный метод: □- однократное использование инвариантных частей алгоритма, при этом реализация изменяющегося поведения остается на усмотрение подклассов; □- необходимость вычленить и локализовать в одном классе поведение, общее для всех

подклассов, чтобы избежать дублирования кода. Это хороший пример техники «вынесения за скобки с целью обобщения», описанной в работе Уильяма Опдайка (William Opdyke) и Ральфа Джонсона (Ralph Johnson) [OJ93]. Сначала выявляются различия в существующем коде, которые затем выносятся в отдельные операции. В конечном итоге различающиеся фрагменты кода заменяются шаблонным методом, из которого вызываются новые операции; □- управление расширениями подклассов. Шаблонный метод можно определить так, что он будет вызывать операции-зацепки (hooks) — см. раздел «Результаты» — в определенных точках, разрешив тем самым расширение только в этих точках.

Шаблонные методы — один из фундаментальных приемов повторного использования кода. Они играют особенно важную роль в библиотеках классов, поскольку предоставляют возможность вынести общее поведение в библиотечные классы.

Шаблонные методы приводят к инвертированной структуре кода, которая подразумевает, что родительский класс вызывает операции подкласса, а не наоборот.

3 Технологическая часть

3.1 Анализ СУБД

Взять в рассмотрение хотя бы 3 из них:

3.1.1 SQLite?

3.1.2 MySQL

3.1.3 Oracle

3.1.4 Microsoft SQL Server

3.1.5 PostgreSQL

3.1.6 MongoDB

3.1.7 Redis

3.1.8 Выбор СУБД для решения задачи

PostgreSQL — основное хранилище, Redis — кеш данных текущей сессии (по окончании сессии перекидывается на диск на стороне сервера (после возврата из функции обработки подключения, т.е. на стороне сервера надо разделять (идентифицировать) конкретные сессии (не пользователей!))).

3.2 Средства реализации

Что на чём написано, как, куда и с помощью чего деплоилось (упомянуть Docker и контейнеризацию). Если в конечном варианте останется Envoy, то рассказать, зачем он нужен.

Golang + Python + SolidJS/React

Внешние средства для мониторинга состояния системы: Docker Desktop (или UI сервиса, на котором развёрнуто приложение), Grafana, PGAdmin.

3.3 Детали реализации

3.3.1 Триггеры БД?

3.3.2 Роли БД

3.3.3 Примеры работы ПО

4 Экспериментально-исследовательская часть

4.1 Цель проводимых измерений

4.2 Описание проводимых измерений

Измеряем врем операций сохранения терминов в БД и поиска терминов по тегам (INSERT и SELECT) на примере нескольких текстов с возрастающим количеством терминов (желательно кратным 10^n).

4.3 Инструменты измерения времени работы программы

4.4 Результаты проведённых измерений

Заключение

В рамках научно-исследовательской работы была проведена классификация сетевых операционных систем для устройств интернета вещей.

В результате сравнения были выделены:

- Azure Sphere, Windows 10 IoT и Amazon FreeRTOS как наиболее функциональные и масштабируемые;
- ОСРВ МАКС и KasperskyOS как наиболее доступные с точки зрения использования прикладных служб;
- Ubuntu Core и Raspbian как наиболее адаптированные для бытового применения.

В ходе выполнения данной работы были решены следующие задачи.

1. Проведён анализ предметной области интернета вещей;
2. Рассмотрены существующие операционные системы для устройств интернета вещей;
3. Сформулированы критерии сравнения и оценки рассмотренных операционных систем;
4. Проведён сравнительный анализ существующих решений по выделенным критериям.

Таким образом, все поставленные задачи были выполнены, поставленная цель достигнута.

Список использованных источников

1. Когаловский М.Р. Энциклопедия технологий баз данных. – Москва: Финансы и статистика, 2002. – 800 с.

2. Алферова Т.К., Леонов А.В., Филиппов К.В. О состоянии автоматизированной базы данных терминов и определений в области ОП // Компетентность. – 2014.– № 7(118). – С. 10-14.

3. Горбач Т.А., Грибова В.В., Окунь Д.Б., Петряева М.В., Шалфеева Е.А., Шахгельдян К.И. База терминов нейрохирургии для интеллектуальной обработки биомедицинских данных // Сборник материалов XIII международной научной конференции: «Системный анализ в медицине». – Благовещенск, 2019. – С. 82-85.

4. Кузнецов И.О. Автоматическое извлечение двусловных терминов по тематике «Нанотехнологии в медицине» на основе корпусных данных // Научно-техническая информация. Сер. 2. – 2013. – № 5. – С. 25-33.

5. Becerro F. B. Phraseological variations in medical-pharmaceutical terminology and its applications for English and German into Spanish translations // SciMedicine Journal. – 2020. – № 2(1). – P.22-29. DOI: 10.28991/SciMedJ-2020-0201-4.

6. Simon N. I., Kešelj V. August. Automatic term extraction in technical domain using part-of-speech and common-word features // Proceedings of the ACM Symposium on Document Engineering. – 2018. – P. 1-4. DOI:10.1145/3209280.3229100.

7. Клышинский Э.С., Кочеткова Н.А., Карпик О.В. Метод выделения коллокаций с использованием степенного показателя в распределении Ципфа // Новые информационные технологии в автоматизированных системах. – 2018. – № 21. – С. 220-225.

8. Кочеткова Н.А. Метод извлечения технических терминов с использованием усовершенствованной меры странности // Научно-техническая информация. Сер. 2.– 2015. –№ 5. – С. 25-32;

Kochetkova N.A. A Method for Extracting Technical Terms Using the Modified Weirdness Measure // Automatic Documentation and Mathematical Linguistics. – 2015 – Vol. 49, № 3. – P. 89-95.

9. Захаров В.П., Хохлова М.В. Автоматическое извлечение терминов из специальных текстов с использованием дистрибутивно-статистического метода как инструмент создания тезаурусов // Структурная и прикладная лингвистика. – 2012. – № 9. – С. 222-233.

10. Terryn A., Hoste V., Lefever E. In no uncertain terms: a dataset for monolingual and multilingual automatic term extraction from comparable corpora // Language Resources and Evaluation. – 2020. – Vol. 54, № 2. – P. 385-418. DOI:10.1007/s10579-019-09453-9.

11. Бутенко Ю.И. Строганов Ю.В., Сапожков А.М. Метод извлечения русскоязычных многокомпонентных терминов в корпусе научно-технических текстов // Прикладная информатика. – 2021. – № 6. – С. 21-27. DOI: 10.37791/2687-0649-2021-16-6-21-27.

12. Бутенко Ю.И. Строганов Ю.В., Сапожков А.М. Система извлечения многокомпонентных терминов и их переводных эквивалентов из параллельных научно-технических текстов // НТИ. Сер. 2. ИНФОРМ. ПРОЦЕССЫ И СИСТЕМЫ. – 2022. – № 9. – С. 12-21. ISSN 0548-0027.

13. К. Дж. Дейт SQL и реляционная теория. Как грамотно писать код на SQL. – Пер. с англ. – СПб.: Символ-Плюс, 2010. – 480 с., ил. ISBN 978-5-93286-173-8

14. Дейт, К. Дж. Введение в системы баз данных, 8-е издание.: Пер. с англ. — М.: Издательский дом ”Вильяме 2005. — 1328 с.: ил. — Парал. тит. англ. ISBN 5-8459-0788-8 (рус.)

15. Маркин, А. В. Системы графовых баз данных. Neo4j : учебное пособие для вузов / А. В. Маркин. — Москва : Издательство Юрайт, 2021. — 178 с. — (Высшее образование). ISBN 978-5-534-13996-9

16. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Паттерны объектно-ориентированного проектирования. — СПб.: Питер, 2020. — 448 с.: ил. — (Серия «Библиотека программиста»). ISBN 978-5-4461-1595-2

Приложение А

Рисунок 1 – Функциональная схема работы системы извлечения многокомпонентных терминов, верхний уровень

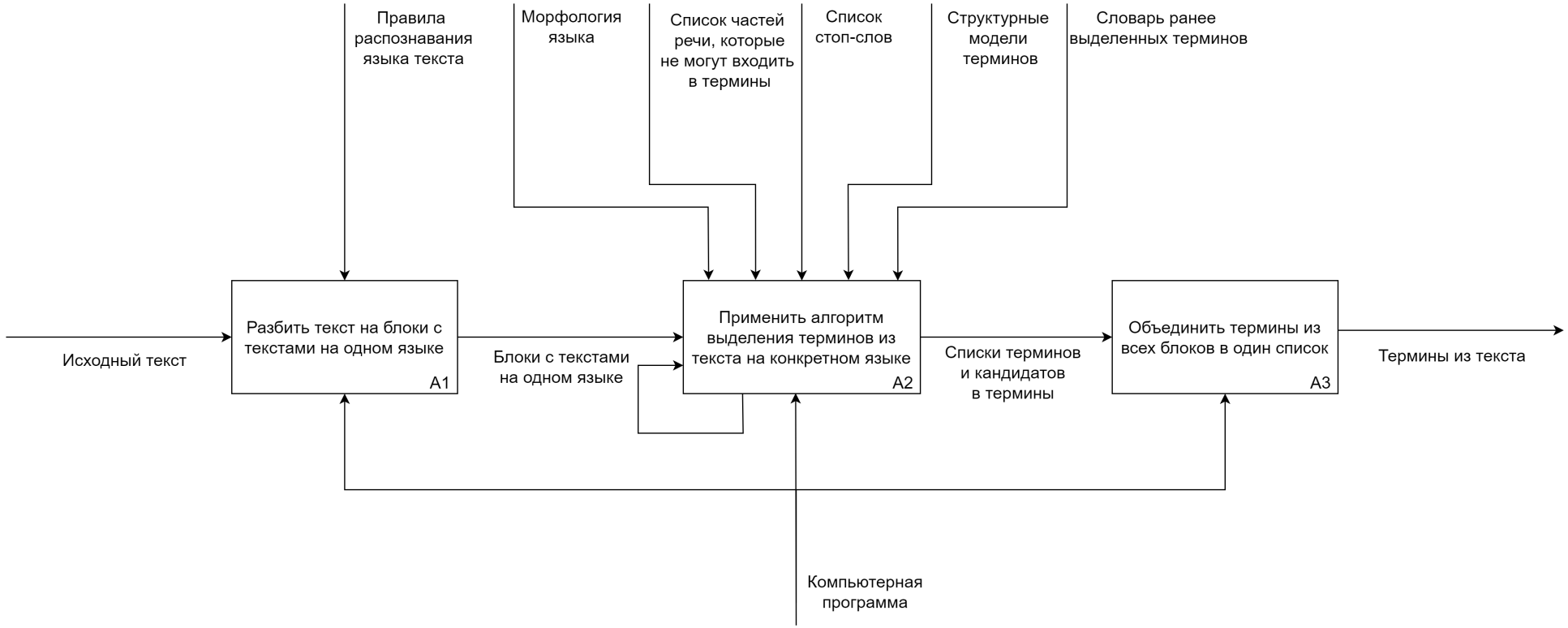

Рисунок 2 – Функциональная схема работы системы извлечения многокомпонентных терминов, декомпозиция уровня A0

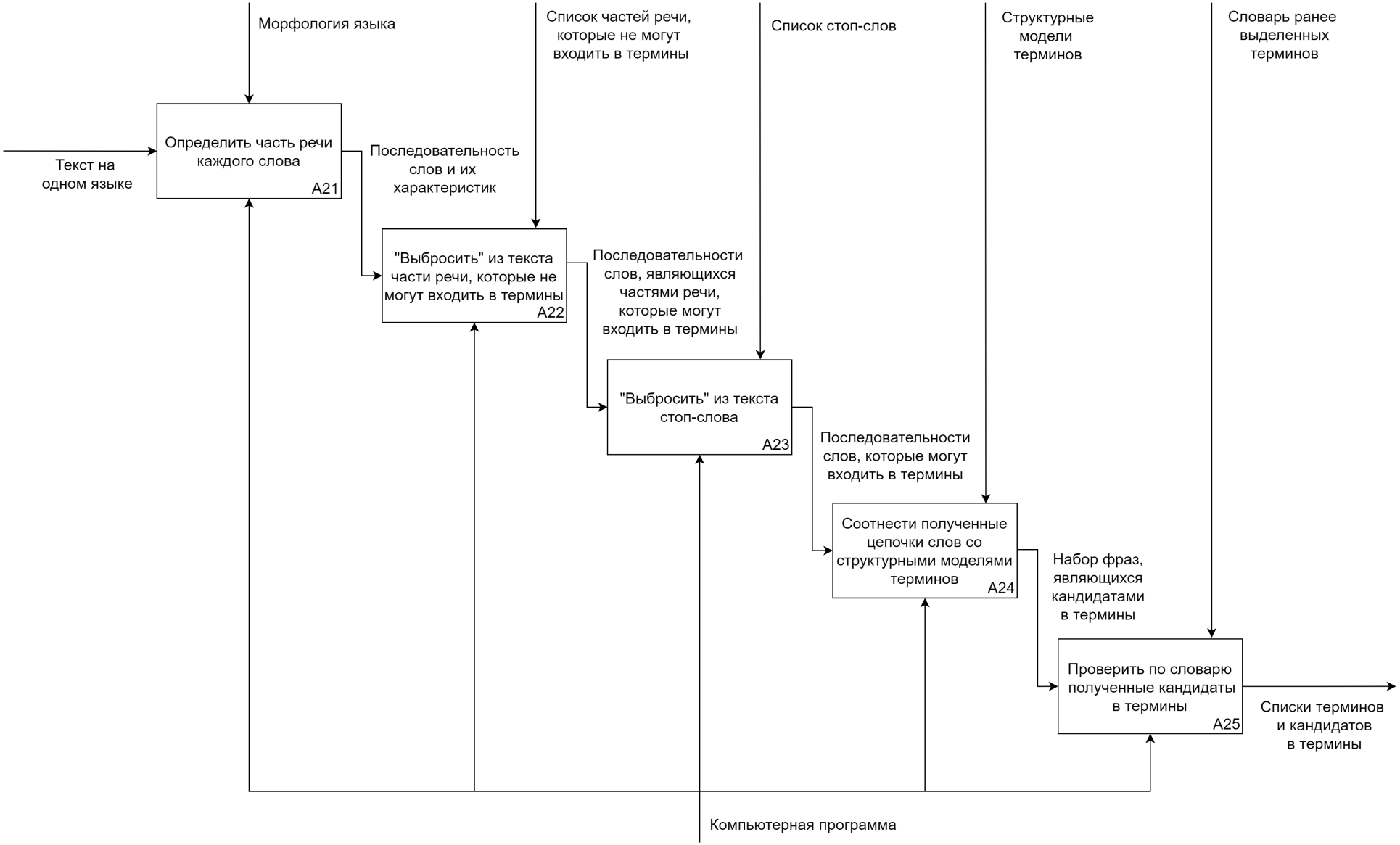

Рисунок 3 – Функциональная схема работы системы извлечения многокомпонентных терминов, декомпозиция уровня A2

Приложение Б

Презентация.