МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ
Кафедра информационных систем управления

Исаченко
Дмитрий Александрович

CROSS-LANGUAGE ФУНКЦИОНАЛЬНОСТЬ АВТОМАТИЧЕСКОГО ПОИСКА В СЕТИ INTERNET РЕЛЕВАНТНЫХ ДОКУМЕНТОВ

Дипломная работа

Научный руководитель**:**
доктор технических наук,
профессор И.В. Совпель

Рецензент:
доктор технических наук,
гл.н.с. ГНУ «ОИПИ НАН БЕЛАРУСИ»
С.Ф. Липницкий

Допущена к защите

«___» _____ 2017 г.

Зав. кафедрой информационных систем управления
доктор технических наук, профессор В. В. Краснопрошин

Минск, 2017

РЕФЕРАТ

Дипломная работа, 51 с., 14 рис., 6 табл., 11 источников.

Ключевые слова: ПОИСК, АВТОМАТИЧЕСКАЯ ОБРАБОТКА ТЕКСТОВ, ПЕРЕВОД, РАЗРЕШЕНИЕ ЛЕКСИЧЕСКОЙ МНОГОЗНАЧНОСТИ, МАШИННОЕ ОБУЧЕНИЕ, ТЕЗАУРУС.

Объект исследования – алгоритмы извлечения ключевой информации из документа, алгоритмы разрешения лексической многозначности при переводе.

Цель работы – исследование методов поиска релевантных документов в многоязычной информационной среде, разработка мобильного клиента для поиска актуальных заданному документов, в том числе предоставленных на языке отличном от языка заданного документа.

Методы исследования – методы теории вероятности, математической статистики, интеллектуальный анализ данных, машинное обучение.

Результатом является выполненный обзор существующих решений в области поиска релевантных документов в многоязычной информационной среде, разработанное мобильное приложение под ОС Android, позволяющее для текстового документа/веб-страницы выполнить поиск релевантной информации в сети Internet, в том числе предоставленной на языке отличном от языка заданного текстового документа/веб-страницы.

Область применения: информационный поиск.

РЭФЕРАТ

Дыпломная работа, 51 с., 14 мал., 6 табл., 11 крыніц.

Ключавыя словы: ПОШУК, АЎТАМАТЫЧНАЯ АПРАЦОЎКА ТЭКСТУ, ПЕРАКЛАД, ДАЗВОЛ ЛЕКСІЧНАЙ ШМАТЗНАЧНАСЦІ, МАШЫННАЕ НАВУЧАННЕ, ТЭЗАЎРУС.

Аб'ект даследавання - алгарытмы здабывання ключавой інфармацыі з дакумента, алгарытмы дазволу лексічнай шматзначнасці пры перакладзе.

Мэта работы - даследаванне метадаў пошуку рэлевантных дакументаў у шматмоўным інфармацыйным асяроддзе, распрацоўка мабільнага кліента для пошуку актуальных дадзенаму дакументаў, у тым ліку на выдатных ад дадзенага дакумента мовах.

Метады даследавання - метады тэорыі верагоднасці, матэматычнай статыстыкі, інтэлектуальны аналіз дадзеных, машыннае навучанне.

Вынікам з'яўляецца выкананы агляд існуючых рашэнняў у вобласці пошуку рэлевантных дакументаў у шматмоўным інфармацыйным асяроддзі, распрацаванае мабільнае прыкладанне пад АС Android, якое дазваляе для тэкставага дакумента/вэб-старонкі выканаць пошук рэлевантнай інфармацыі ў сетцы Internet, у тым ліку прадстаўленай на мове выдатнай ад мовы дадзенага тэкставага дакумента/вэб-старонкі.

Вобласць прымянення: інфармацыйны пошук.

ABSTRACT

Diploma thesis, 51 p., 14 fig., 6 tabl., 11 references.

Keywords: SEARCH, AUTOMATIC TEXT PROCESSING, TRANSLATION, WORD-SENSE DISAMBIGUATION, MACHINE LEARNING, THESAURUS.

The object of research - algorithms of key information extraction, algorithms solving word-sense disambiguation problem during translation.

The objective is to study various methods of searching for relevant documents in a multilingual information environment, development of a mobile client for searching relevant documents, including those whose language are different from the language of source document.

The methods of research - methods of probability theory, mathematical statistics, data mining, machine learning.

The result is review of existing solutions in the field of searching for relevant documents in a multilingual information environment, developed mobile application for the Android OS, which allows for a text document/webpage to search for relevant information on the Internet, including those whose language are different from the language of the source text document/webpage.

The application area: information retrieval.

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ И СИМВОЛОВ

ЕЯ - естественный язык;

ПОД - поисковой образ документа;

Стоп-слова - слова, не несущие в себе смысловой и содержательной нагрузки, такие как междометия, предлоги и прочие;

Стемминг - процесс нахождения основы слова для заданного исходного слова;

TF– частота термина в контексте определённого документа;

IDF - обратная частота термина в корпусе документов;

IR – информационный поиск;

CLIR - разновидность информационного поиска, при котором язык извлечённой информации может отличаться от языка исходного запроса;

WSD - проблема, связанная с разрешением лексической многозначности.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

В наше время огромное количество информации доступно в электронном виде. Информационные системы, оперирующие большими объемами данных произвольной предметной области и успешно решающие различные прикладные задачи, становятся все более востребованными, как предприятиями и организациями, так и отдельными пользователями.

Информационный поиск(IR) представляет собой процесс извлечения релевантной информации среди огромного количества документов. Традиционные IR системы реализуются в основном для документов, написанных на одном языке, хотя интернет сам по себе является многоязычной информационной средой. По этой причине возникает языковой барьер между пользователем и доступной информацией, а также появляется необходимость в исследовании и разработке методов для повышения эффективности IR.

В большинстве случаев при поиске информации в интернете мы хотим, чтобы она была написана на нашем родном языке, однако такая информация не всегда является доступной. С учётом того, что большинство пользователей владеет одним или несколькими иностранными языками, они могут быть также заинтересованы в поиске информации, предоставленной на других языках. Так появляется необходимость в многоязычном поиске(CLIR), целью которого является сопоставления запроса, написанного на одном языке, с документами, написанными на других языках. CLIR снимает языковой барьер, благодаря чему пользователи могут отправлять запросы, написанные на их родном языке, а получать документы на других языках и наоборот. Например, запрос на русском языке вернёт релевантную информацию на английском языке. Из-за быстрого развития интернет-технологий потребность в CLIR значительно растёт, поскольку данный тип поиска позволяет реализовать обмен информацией между различными языками, устранить лингвистическое несоответствие между предоставляемыми запросами и документами, которые извлекаются из информационной сети. В связи с этим CLIR приобрел большое значение, как в качестве исследовательской дисциплины, так и в качестве технологии, которая будет востребована на рынке.

В дополнение к проблемам, встречаемым при одноязычном IR, в CLIR добавляется ещё одна – проблема перевода. Однако в данном случае перевод будет отличаться от полнотекстового машинного перевода. Причиной этому является отсутствие необходимости быть удобочитаемым для человека, перевод должен просто максимально подходить для поиска соответствующих документов. В основе CLIR могут лежать следующие варианты реализации перевода: перевод запроса, перевод документов, перевод запроса и документов одновременно. Уже было опубликовано большое количество исследований на

тему реализации CLIR. Многие вопросы, связанные с данной темой, также рассматриваются на различных конференциях, например, TREC, NTCIR, CLEF. Каждая из данных конференций охватывает определённые языки: TREC включает в рассмотрение испанский, китайский, немецкий, французский, арабский и итальянский языки; NTCIR включает японский, китайский и корейский языки, а CLEF - французский, немецкий, итальянский, испанский, голландский, финский, шведский и русский.

В дипломной работе сначала приводится описание подходов реализации поиска релевантных документов в одноязычной информационной среде. Затем выполняется анализ техник перевода, а также методов разрешения лексической многозначности для осуществления поиска в многоязычной информационной среде. Итогом проведенного в дипломной работе исследования является разработанное мобильное приложение, обладающее cross-language функциональностью при поиске релевантных документов в сети интернет.

ГЛАВА 1
ИССЛЕДОВАНИЕ ПОДХОДОВ РЕАЛИЗАЦИИ ПОИСКА В СЕТИ ИНТЕРНЕТ РЕЛЕВАНТНЫХ ДОКУМЕНТОВ

Задача автоматизации поиска в сети интернет документов релевантных данному относится к классическим задачам информационного поиска и её можно решать одним из двух следующих способов:

- разработать собственную поисковую систему;
- для заданного документа составить ПОД, который будет представлять собой запрос для уже существующих поисковых систем.

1.1 Структурно-функциональная схема информационно-поисковой системы

Работу поисковой системы можно представить следующим образом:

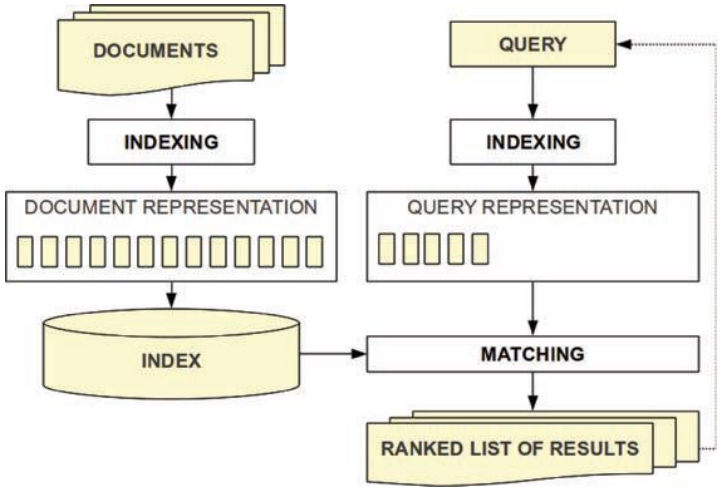

Рисунок 1.1 – Схема работы поисковой системы

Основными её составляющими являются: поисковый робот, индексатор, поисковик.

Поисковый робот — составная часть поисковой системы, основной функцией которой является перебор страниц Интернета. Данный перебор осуществляется для сохранения информации о страницах в базе данных поисковика. Поисковой робот исследует содержимое страницы и затем сохраняет поисковой образ на сервере поисковой машины, которой принадлежит, после этого исследуются следующие страницы, которые доступны по ссылкам с текущей. Большие сайты зачастую проиндексированы поисковой машиной не целиком, так как обычно для поисковой машины глубина проникновения внутрь сайта и максимальный размер сканируемого текста

ограничены. Переходы между страницами реализуются с помощью ссылок, которые содержатся на исходных страницах. В зависимости от алгоритмов информационного поиска определяются порядок обхода страниц и частота визитов, так же предотвращается возможность зацикливания.

Современны поисковые системы с целью ускорения процесса индексирования сайта дают пользователю возможность ручного добавления сайта в очередь для индексирования. Если на сайт невозможно попасть по внешним ссылкам, то это вообще оказывается единственной возможностью уведомить поисковую систему о существовании сайта.

В ходе процесса индексирования робот поисковой системы помещает в базу данных сведения о сайте (ключевые для сайта слова, ссылки, изображения, аудио…), которые затем используются при поиске. Индексирование страницы осуществляется непосредственно с помощью индексатора, в обязанности которого входит анализ страницы, при этом каждый элемент веб-страницы анализируется отдельно. Полученные индексатором данные о веб-страницах помещаются в индексную базу данных для возможности использования их в последующих запросах.

Поисковый запрос - последовательность символов, которую пользователь вводит в поисковую строку, для обнаружения релевантной информации. Формат поискового запроса зависит от 2-х вещей: от типа информации для поиска и от устройства поисковой системы. Обычно поисковой запрос представляет собой набор слов или фразу.

Работу поисковой системы можно разбить на следующие шаги: сначала исходный контент принимается поисковым роботом, затем согласно контенту, в ходе процесса индексирования определяется доступный для поиска индекс, после чего можно обнаруживать с помощью поисковой системы исходные данные. Данные шаги выполняются каждый раз при обновлении поисковой системы.

В большинстве случаев для поисковых систем основным источником для анализа и получения информации о веб-странице является HTML страница, соответствующая ей. Основное внимание при извлечении информации уделяется заголовкам и метатегам. Поисковые гиганты, такие как Google, имеют возможность полностью сохранять контент исходной страницы целиком или только часть его(кэш). Последнее позволяется значительно увеличить скорость поиска информации на ранее посещённых страницах(кэшированные). Текст запроса пользователя обычно сохраняется вместе с кэшированной страницей, чтобы сохранить актуальность в случае обновления исходной. Пользователь формирует запросы для поисковика, который затем обрабатывает их, анализируя данные полученные в ходе процесса индексации, и затем возвращает результаты поиска. Запросы пользователя зачастую представляют собой набор ключевых

слов. В тот момент, когда пользователь вводит запрос, поисковая система уже начинает анализировать имеющиеся индексы, после чего пользователь получает наиболее релевантные для него веб-страницы. Также поисковая системы может возвращать веб-страницы вместе с краткой аннотацией, которая представляет собой заголовок документа и, возможно, некоторый отрывок из текста. Поисковая система характеризуется следующими двумя оценками: оценка точности найденных релевантных страниц и оценка полноты найденных релевантных страниц. Для того, чтобы в начале списка результаты были наиболее актуальными для пользователя, многие поисковые систем используют методы ранжирования, которые в свою очередь посредством определения, какие страницы являются наиболее релевантными для пользователя, формируют очередь отображения результатов.

В связи с огромной трудоёмкостью разработки собственной поисковой системы в данной работе будет использоваться поисковая система Google. Таким образом задача автоматизации поиска в сети интернет документов релевантных данному сведётся к формированию поискового запроса. Поисковым запросом для Google будет являться поисковой образ текстового документа, который формируется из ключевых для исходного текста слов. Согласно рекомендациям Google, поисковой запрос должен состоять из ключевых слов, оптимальное количество которых должно находиться в диапазоне 6-10.

1.2 Поиск релевантных документов в одноязычной информационной среде

1.2.1 Предварительная обработка документа

В ходе предварительной обработки документа происходят следующие действия: токенизация, удаление стоп-слов, стемминг и расширение терминов.

Токенация служит для распознавания и изолирования различных языковых единиц, присутствующих в исходном тексте. Двумя основными процедурами процесса токенизации являются сегментация слов и декомпозиция слов. Сегментация обычно выполняется при работе с восточноазиатскими языками, в то время как декомпозиция с европейскими.

Сегментация - это процесс разбития исходного текста на составляющие единицы. Данный процесс легко реализовать для языков, в которых явно выделены границы слов, например, с помощью пробельного символа в английском и французском языке, но значительно труднее для таких языков, как китайский, где разделители между словами отсутствуют.

Один из подходов реализации сегментации использует алгоритм максимального соответствия, в основе которого лежит список известных слов.

Очевидно, что такой подход не работает для слов, которые отсутствуют в исходном списке. Альтернативой данному подходу являются подходы, основанные на n-граммах, наиболее распространёнными из которых являются подходы, использующие биграммы.

В некоторых языках, таких как русский и немецкий, часто употребляются сложные слова, которые состоят из нескольких слов и в ходе процесса токенизации должны считаться одной языковой единицей. Для обнаружения сложных слов можно использовать специальные словари, содержащие их список. Текст будет разбит на минимальное количество слов, присутствующих в данном словаре. Если алгоритм обнаружил два (или более) возможных вариантов составного слова в определённом отрывке текста, то должен выбраться наиболее вероятный для данного контекста. Вероятность можно вычислять, предварительно обучив систему на корпусе документов.

Предлоги, местоимения, союзы, общие глаголы и незначащие слова обычно удаляются из исходного текста до составления ПОД. Фильтрация этих терминов осуществляется зачастую с использованием списка стоп-слов.

Список стоп слов для английского языка:

a, about, above, after, again, against, all, am, an, and, any, are, aren't, as, at, be, because, been, before, being, below ,between, both, but, by, can't, cannot, could, couldn't, did, didn't, do, does, doesn't, doing, don't, down, during, each, few, for, from, further, had, hadn't, has, hasn't, have, haven't, having, he, he'd, he'll, he's, her, here, here's, hers, herself, him, himself, his, how, how's, i, i'd, i'll, i'm, i've, if, in, into, is, isn't, it, it's, its, itself, let's, me, more, most, mustn't, my, myself, no, nor, not, of, off, on, once, only, or, other, ought, our, ours, ourselves, out, over, own, same, shan't, she, she'd, she'll, she's, should, shouldn't, so, some, such, than, that, that's, the, their, theirs, them, themselves, then, there, there's, , these, they, hey'd, they'll, they're, they've, this, those, through, to, too, under, until, up, very, was, wasn't, we, we'd, we'll, we're, we've, were, weren't, what, what's, when, when's, where, where's, which, while, who, who's, whom, why, why's, with, won't, would, wouldn't, you, you'd, you'll, you're, you've, your, yours, yourself, yourselves.

Одним из способов повышения эффективности работы систем информационного поиска является предоставление поисковым системам возможности обнаружения различных форм одного и того же слова посредством стемминга. Стемминг представляет собой морфологический разбор слова, в ходе которого обнаруживается общая для всех его грамматических форм основа, обрубаются окончания и суффиксы.

Существует несколько критериев оценки стеммеров: корректность, эффективность поиска и производительность сжатия.

При реализации стемминга нужно найти баланс между следующими двумя проблемами: чрезмерный стемминг, приводящий к объединению несвязанных

терминов и соответственно понижающий точность поиска, так как извлекаются нерелевантные документы; основа слова выделяется слишком слабо, в связи с чем будет понижаться полнота поиска.

Для английского языка на данный момент самым популярным является стеммер Портера [1] в силу его быстрой скорости работы, отсутствия необходимости в предварительной обработке корпуса документов и использования каких-либо баз основ. В основе данного стеммера лежит алгоритм усечения окончаний, использующий для своей работы небольшой набор правил, например, если слово оканчивается на “ет”, то удалить “ет” и так далее. Алгоритмы усечения окончаний достаточно эффективны на практике, но в то же время обладают некоторыми недостатками. Алгоритмы усечения окончаний неэффективны в случае изменения корня слова, например, изменение или выпадение гласной. Данные алгоритмы эффективны для тех частей речи, которые имеют хорошо известные окончания и суффиксы. Стеммер Портера основывается на том, что количество словообразующих суффиксов в языках ограниченно. Благодаря этому алгоритм может выполняться с помощью установленных вручную определённых правил. Алгоритм выделения основы слова стеммера Портера для английского языка включает в себя пять шагов, на каждом из которых проверяется будет ли получившаяся в результате убирания словообразующего суффикса часть соответствовать заранее установленным правилам. В случае, если правила удовлетворены осуществляется переход на следующий шаг алгоритма, иначе выбирается другой суффикс для отсечения. Из описания хода работы алгоритма видно, что у стеммера Портера существует недостаток: он может обрезать слово больше необходимого, что в свою очередь затруднит получение правильной основы слова и соответственно уменьшит точность извлечения релевантной информации. Ещё одним недостатком стеммера Портера является отсутствие возможности работать при изменении корня слова, например, в случае выпадающих беглых гласных.

Другой разновидностью стеммеров являются стеммеры, использующие таблицы поиска флективных норм. Трудностью при реализации данного типа стеммеров является необходимость перечисления всех флективных форм в таблице. Если какая-то из форм отсутствует, то она обрабатываться не будет. В связи с этим получается, что таблица поиска может иметь большой размер. В качестве плюсов можно выделить простоту подхода, скорость работы и простоту обработки исключений. Таблицы поиска, которые используются в стеммерах, обычно генерируются в полуавтоматическом режиме. Чтобы избежать проблемы, когда разные слова относятся к одной лемме (ошибка лемматизации), при реализации алгоритма поиска можно использовать предварительную частеречную разметку. Это позволит применять соответствующие для каждой части речи правила нормализации.

Основной недостаток классических стеммеров – они не различают слова, имеющие схожий синтаксис, но абсолютно разные значения, например, в английском языке “news” и “new” для данных стеммеров будут различными формами одного и того же слова. С целью разрешения этой проблемы были реализованы стеммеры на основе корпусов текстов. Ключевой идеей данных стеммеров является создание классов эквивалентности для слов классических стеммеров, которые после разделят некоторые объединенные слова, анализируя их встречаемости в корпусе. Для определения основы слова алгоритм сопоставляет его с основами из базы данных, используя различные ограничения, такие как длина искомой основы в слове относительно длины самого слова и т.п.

1.2.2 Составление поискового образа документа

Поисковый образ документа(ПОД) - текст, выражающий на информационно поисковом языке основное содержание документа и в последующем используемый для информационного поиска. Для формирования ПОД необходимо выделить из документа ключевую информацию.

Любой алгоритм извлечения ключевых слов/словосочетаний реализует одну или несколько систем распознавания образов, разбивающих входное множество слов на два класса: ключевые и прочие. По наличию элементов обучения выделяют необучаемые, обучаемые и самообучаемые методы извлечения ключевых слов. Более простые необучаемые методы подразумевают контекстно-независимое выделение ключевых слов/словосочетаний из отдельного текста на основе априорно составленных моделей и правил. Они подходят для гомогенных по функциональному стилю корпусов текстов, увеличивающихся со временем в объемах, например, научных работ или нормативных актов. Обучаемые методы предполагают использование разнообразных лингвистических ресурсов для настройки критериев принятия решений при распознавании ключевых слов. Здесь большое значение имеет корректное выделение ключевых слов в выборке, используемой для обучения. Среди методов с обучением можно выделить подкласс самообучаемых, если обучение ведется без учителя или с подкреплением (на основе пассивной адаптации). По второму признаку классификации, прежде всего, следует выделить статистические и структурные методы извлечения ключевых слов. Статистические методы учитывают относительные частоты встречаемости морфологических, лексических, синтаксических единиц и их комбинаций. Это делает создаваемые на их основе алгоритмы довольно простыми, но недостаточно точными, т.к. признак частотности ключевых слов не является превалирующим.

Для извлечения ключевых словосочетаний из текста выполняется анализ коллокаций. Коллокация состоит из нескольких слов, представляющих собой синтаксически и семантически целостную единицу. При извлечении коллокаций анализируют является ли появление лексических единиц случайным или нет.

В нашем случае ПОД, будет состоять из ключевых слов исходного документа и являться запросом для поисковой системы Google.

1.2.3 Анализ методов извлечения ключевых слов

Существуют следующие категории методов извлечения ключевых слов: статистические, лингвистические и гибридные, которые являются комбинацией первых двух [2].

В основе лингвистических методов лежат значения слов, семантические данные о слове, а также используются онтологии, которые формализуют знания из некоторой области с помощью концептуальной схемы. При использовании данных подходов возникает трудность, связанная с реализацией онтологий, что само по себе является очень трудоёмким процессом. Часть операций, которая при лингвистическом анализе текстов выполняется вручную, усложняют процесс анализа документов из-за дополнительной возможности возникновения ошибок и неточностей.

Наиболее популярными лингвистическими методами при обработке естественного языка являются лингвистические методы в основе которых лежат графы. Главная задача данных методов представляет собой построение семантического графа. Семантический граф является взвешенным графом. Термины исходного документа будут вершинами в графе. Между вершинами графа есть ребро в том и только в том случае, если присутствует семантическая связь между терминами. Вес в семантическом графе равен значению семантической близости связанных ребром терминов. Поиск ключевых слов осуществляется с помощью алгоритмов обработки графа. Определяющими характеристиками лингвистических методов, основанных на графах, являются способ отбора множества терминов, а также алгоритм определения весов рёбер (семантической близости терминов).

Статистические методы базируются на численных данных о встречаемости слова в тексте [3]. Основными их преимуществами являются относительная простота реализации, универсальность алгоритмов поиска ключевых слов, а также отсутствие необходимости в выполнении трудоемких операций построения лингвистических баз знаний. Максимальную точность и полноту имеют алгоритмы, в основе которых лежат статистические исследования корпусов документов. Алгоритмы, которые предварительно не обрабатывают никаких документов, кроме того, ключевые слова которого необходимо извлечь,

обладают сравнительно более низкой точностью. Классическими подходами в области статистической обработки естественного языка можно считать использование метрики TF-IDF и ее модификаций при поиске ключевых слов, а также анализ коллокаций при поиске ключевых словосочетаний. Одним из самых простых статистических методов выделения ключевых слов в тексте является построение множества кандидатов путем ранжирования по частоте встречаемости в исходном документе всех его словоформ или лексем. Фильтрация в данном случае осуществляется через отбор в качестве ключевых наиболее частотных словоформ/лексем.

При реализации статистических подходов для поиска ключевых слов задействованы различные эвристические алгоритмы, обычно приводящие словоформу к ее квази-основе, что достигается посредством выделения у словоформы некоторого количества букв. Данные алгоритмы (стемминг-алгоритмы) обсуждались выше в описании предварительной обработки документа. В ходе алгоритмов стемминга выделяются основы слов, которые затем ранжируются по частоте. Словоформы с наибольшей частотой считаются ключевыми. Статистические методы, обученные для повышения точности поиска ключевых слов на корпусе текстов, достаточно популярны. В тоже время необходимо наличие таких корпусов для каждой определённой предметной области, что значительно затрудняет возможность реализации данных методов. С целью повышения точности описания контента документа разрабатываются методики, у которых мерой релевантности является вес лексемы, полученный посредством определённой комбинации значений различных параметров лексем, таких как, расположения в тексте, статистика совместной принадлежности слов одному и тому же документу и т.п.

Положительными сторонами использования статистических методов является универсальность и относительная простота реализации алгоритмов извлечения ключевых слов, которая связана с отсутствием необходимости выполнять трудоемкие и занимающие огромное количество времени операции для создания лингвистических баз знаний. Однако методы выделения ключевых слов, в основе которых лежит только статистический подход иногда не обеспечивают желаемого качества результатов, особенно невысокие результаты получаются при работе с языками с богатой морфологией, например, с русским языком, в котором лексемы характеризуются огромным количеством словоформ с невысокой частотностью в отдельно рассматриваемом тексте.

Для оценки важности слова в контексте документа будет рассмотрена более подробно статистическая мера TF-IDF, которая является произведением двух статистик: частоты термина в данном документе и обратной частоты термина в корпусе документов. Существуют различные способы определения данных статистик.

Будут введены следующие обозначения:

1. D - корпус документов;
2. N - размер корпуса документов;
3. t - термин, важность которого хотим определить в документе d;
4. n(t) = 1 + количество документов, в которых встречается t;
5. f(t,d) - частота термина t в документе d.

Способы определения статистики TF:

- по частоте встречаемости (raw frequency) формула (1.1);
- логический (boolean frequency) формула (1.2);
- логарифмически нормализованный (logarithmically scaled frequency) формула (1.3);
- нормализованный по максимальной частоте слова (augmented frequency) формула (1.4).

$$tf(t,d) = f(t,d) \tag{1.1}$$

$$tf(t,d) = \begin{cases} 1, & f(t,d) > 0 \\ 0, & f(t,d) = 0 \end{cases} \tag{1.2}$$

$$tf(t,d) = \begin{cases} 1 + \log(f(t,d)), & f(t,d) > 0 \\ 0, & f(t,d) = 0 \end{cases} \tag{1.3}$$

$$tf(t,d) = 0.5 + \frac{0.5 * f(t,d)}{\max\{f(t',d) : t' \in d\}} \tag{1.4}$$

Способы определения статистики IDF представлены формулами (1.5) - (1.8).

$$idf(t,D) = 1 \tag{1.5}$$

$$idf(t,D) = \log\left(\frac{N}{1+n(t)}\right) \tag{1.6}$$

$$idf(t,D) = \log\left(\frac{\max\{n(t'), t' \in d\}}{1+n(t)}\right) \tag{1.7}$$

$$idf(t,D) = \log\left(\frac{N-n(t)}{n(t)}\right) \tag{1.8}$$

Различные варианты схемы взвешивания TF-IDF часто используются поисковыми системами в качестве основного инструмента при ранжировании по релевантности документов для данного поискового запроса. Так же TF-IDF может быть успешно использован при фильтрации стоп-слов в различных

предметных областях.

Для повышения точности автоматического обнаружения ключевых слов в тексте используются гибридные методы, представляющие собой комбинацию статистических методов обработки документов, дополненных несколькими лингвистическими процедурами, такими как морфологический, синтаксический, и семантический анализ, а также различными лингвистическими базами знаний. В основе гибридных методов поиска ключевых слов в документе, может лежать обучение на корпусе текстов. Например, метод Кена Баркера, осуществляет поиск в исходном тексте базовых именных групп(БИГ) посредством морфосинтаксического анализа с использованием словарей и расчётом релевантности БИГ. Именные группы, обладающие показателем релевантности выше заданного порога, относятся к ключевым. Одной из разновидностей гибридных методов поиска ключевых слов являются методы на основе машинного обучения, в которых выделение ключевых слов представляет собой задачу классификации. Как известно, для построения обучающей выборки, по которой будет обучен классификатор, необходимы корпуса документов, в которых выделены ключевые слова. Выделенные ключевые слова играют роль положительного примера, остальные слова – отрицательного примера. После этого для всех слов тренировочного текста вычисляется их релевантность, посредством сопоставления каждого из слов с вектором значений различных параметров. Запоминается разница между значениями векторов данных параметров для ключевых и не являющихся таковыми слов. Затем происходит обучение модели посредством расчёта вероятности принадлежности каждого слова к группе ключевых и задания соответствующего порога. Поиск ключевых слов во входном документе осуществляется с помощью классификатора, путем расчёта актуальности слов в соответствии с построенной моделью.

Проанализировав вышеописанные методы, было замечено, что схема выделения ключевых слов в тексте схожа (см. Рисунок 1.2) для каждого из них и её можно разбить на следующие шаги:

1. Предварительная обработка, представляющая текст в формате удобном для последующего анализа. В неё входят следующие операции: фильтрация из исходного текста стоп-слов, не несущих смысловой нагрузки (предлоги, союзы, частицы, местоимения, междометия и т. д.), выделение основы слова;
2. Отбор кандидатов: выделяются все возможные слова, фразы, термины или понятия (в зависимости от поставленной задачи), которые потенциально могут быть ключевыми;
3. Анализ свойств: для каждого кандидата нужно вычислить свойства, которые указывают, что он может быть ключевым. Например, кандидат, появляющийся в названии книги, скорей всего является ключевым;

4. Отбор ключевых слов из числа кандидатов, посредством вычисления весов важности ключевых слов/словосочетаний в контексте документа.

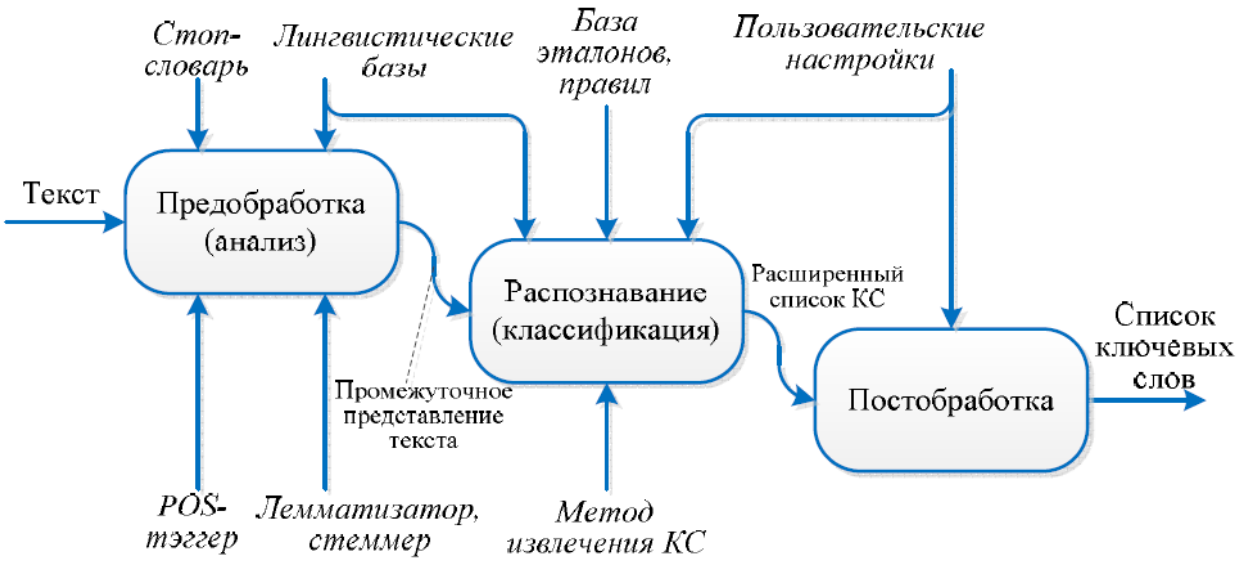

Рисунок 1.2 – Процесс извлечения ключевых слов

В связи с трудоёмкостью реализации собственного лингвистического процессора и недостаточной точностью методов выделения ключевых слов, в основе которых лежит только статистический подход, в данной работе для выделения ключевых слов при составлении ПОД будет использоваться сторонний сервис, использующий гибридный подход для извлечения ключевых слов. Анализ существующих сервисов наиболее популярных в IT сообществе для решения данной задачи будет приведён в следующей главе.

1.3 Поиск релевантных документов в многоязычной информационной среде

При поиске информации в многоязычной информационной среде необходимо сопоставлять запросы и документы, написанные на разных языках. Для разрешения несоответствия языков используется перевод запроса и/или документов перед выполнением поиска [4]. Поэтому правильность перевода одна из главных задач при CLIR.

При разработке собственной поисковой системы, поддерживающей обнаружение информации на языке отличном от языка запроса, можно было бы перевести все имеющиеся документы на все возможные языки запросов (см. Рисунок 1.3).

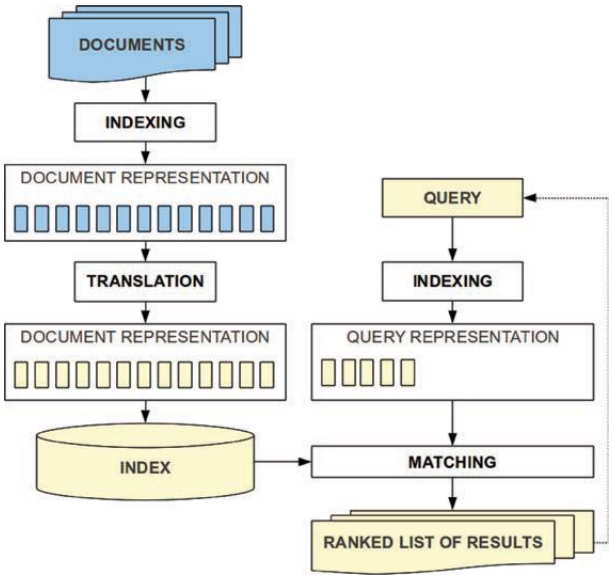

Рисунок 1.3 – CLIR с переводом документов

Данный подход является вычислительно затратным, а также возникает необходимость хранить переводы всех документов системы на всевозможные языки. Следующий подход реализации многоязычного поиска предлагает все документы и поисковой запрос переводить на промежуточный язык (см. Рисунок 1.4).

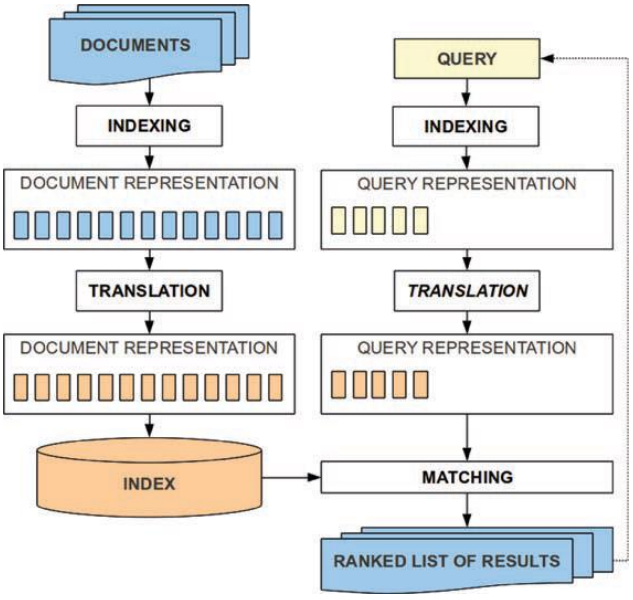

Рисунок 1.4 – CLIR с переводом документов и запроса на промежуточный язык

Так же существует 3-й подход, основанный только на переводе запроса. Данный подход является наименее вычислительно затратным, а также не требует

дополнительного пространства для хранения переведённых документов, являясь в то же время наиболее предпочтительным для реализации с точки зрения CLIR сообщества (см. Рисунок 1.5).

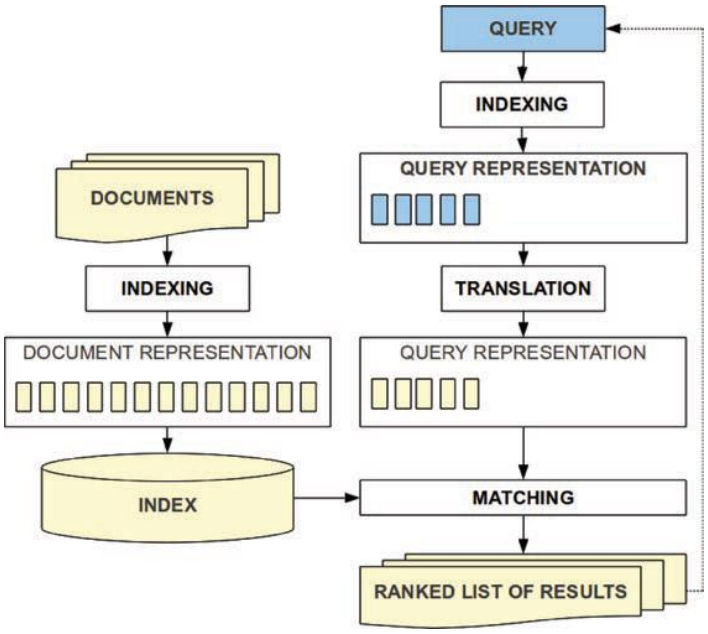

Рисунок 1.5 – CLIR с переводом запроса

Согласно схеме, изображённой на рисунке 1.5, запросом будет является текстовый документ, процесс индексации – процесс построения поискового образа документа.

Сравнительная характеристика 3-х подходов к реализации перевода приведена ниже (см. таблица 1.1).

Таблица 1.1 - Сравнение трех подходов к переводу

Параметры	**Перевод запроса**	**Перевод документа**	**Перевод документа и запроса**
Неоднозначность	Высокая	Низкая	Средняя
Дополнительное пространство для хранения	Не требуется	Необходимо	Не требуется
Время перевода	Низкое	Высокое	Высокое
Поиск информации	Двуязычный	Двуязычный	Двуязычный и многоязычный
Гибкость	Высокая	Низкая	Низкая

Таким образом для исходного текста сначала будет составляться ПОД, а затем выполнять его перевод. При осуществлении перевода появляется проблема разрешения лексической многозначности слов, для решения которой используются различные внешние источники знаний, например, лексические базы данных, тезаурусы. Более подробно разрешения лексической многозначности слов при переводе будет рассмотрено в следующей главе.

1.3.1 Лексические базы данных. WordNet

Тезаурус – словарь, охватывающий понятия, определения и термины некоторой области знаний [5]. Слова в тезаурусах упорядочены по смысловой близости, а не по алфавиту.

Наиболее распространёнными типами смысловых отношений между словами в тезаурусах являются:

- синонимия, базирующаяся на критерии, что два выражения являются синонимичными в том случае, когда замена одного из них на другое в предложении не изменяет смысл данного предложения, например, быстрый – шустрый, бортпроводница – стюардесса;
- антонимия, основанная на смысловом противопоставлении, например, тёплый – холодный, светло – темно;
- гипо-гиперонимия, представляющая собой отношение общего и частного, например, машина – самосвал;
- меронимия, т.е. отношение часть-целое, например, компьютер – процессор, тетрадь – страница.

Синсетом называется множество слов, связанных отношением синонимии. Синсеты разбивают множество всех лексических единиц на классы эквивалентности. Если для некоторого слова не существует синонимов, то соответствующий ему синсет будет состоять только из одного слова. Разные значения многозначных слов входят и в разные синсеты: золотая (монета) – сделанная из золота и золотой (работник) – хороший.

WordNet - это огромная свободно распространяющаяся и соответственно общедоступная для загрузки лексическая база знаний для английского языка. WordNet является семантической сетью, узлы которой представляют собой синсеты, связанные различными отношениями, такими как гипонимия, гиперонимия, голонимия, меронимия и т.п. [6, 7]. WordNet приобрёл популярность благодаря его содержательным и структурными характеристиками. Принстонский WordNet и все последующие варианты для других языков направлены на отображение состава и структуры лексической системы языка в целом, а не отдельных тематических областей. Для каждого синсета имеется описание на естественном языке, а так же примеры использования входящих в него слов. Лексемы, входящие в состав тезауруса,

могут относиться к четырем частям речи: существительное, прилагательное, наречие и глагол. Лексемы различных частей речи хранятся отдельно, и описания, соответствующие каждой части речи, имеют различную структуру.

Синсеты взаимосвязаны между собой посредством концептуально-семантических и лексических отношений. Основным отношением между словами в WordNet является синонимия. Синонимы - слова, которые обозначают одну и ту же концепцию и являются взаимозаменяемыми во многих контекстах. Каждый из 117 000 синсетов WordNet связан с другими синсетами с помощью небольшого числа смысловых отношений. Кроме того, синсет содержит краткое определение и, в большинстве случаев, одно или несколько коротких предложений, иллюстрирующих использование слов из данного синсета. Формы слов с несколькими различными значениями представлены в виде множества различных синсетов. Синонимы обязаны быть взаимозаменяемы хотя бы в некотором непустом множестве контекстов. Для отношения синонимии не требуется заменимость всех синонимов во всех контекстах, в противном случае количество синонимов было бы слишком малым в языках. Существительные в WordNet могут иметь следующие семантические отношения: синонимия, антонимия, гипонимия/гиперонимия, меронимия.

Наиболее часто встречающимся отношением между синсетами является гиперонимия и гипонимия. Гиперонимия связывает более общие синсеты, такие как мебель, с более специфическими, такими как кровать. Таким образом, согласно WordNet в категорию мебели входит кровать, которая, в свою очередь, включает в себя двухъярусную кровать. Наоборот, понятия типа кровати и двухъярусной кровати составляют категорию мебели. Отношение гипонимии является переходным: если кресло является своего рода стулом, а стул есть мебель, то кресло является своего рода мебелью. Синсет А – гипоним синсета В, в том случае, когда существуют предложения типа А есть (является разновидностью) В. И соответственно наоборот синсет А – гипероним синсета В, в том случае, когда существуют предложения типа А имеет разновидность В.

Меронимия или другими словами отношение «часть-целое» имеет место между синсетами, такими как, например, стул и спинка, стул и ножки. В WordNet определены три подвида отношения часть-целое: быть частью, быть элементом, быть сделанным из. Части у различных сущностей могут иметь одинаковое название, например, острие может быть у иголки, карандаша, стрелы, ножа, булавки и т.д. Таким образом А является меронимом В, в том случае, если предложения вида А содержит В и А является частью В естественны для А и В, интерпретируемых как родовые понятия.

Так же в WordNet выделяют 2 категории глаголов согласно их смысловому значению: глаголы, обозначающие действия (действия и события), и глаголы состояния. Среди глаголов действий и событий выделяют следующие 14 групп:

контакта, движения, коммуникации, восприятия, изменения, соревнования, познания, создания, эмоций, потребления, обладания, относящиеся к социальному поведению и глаголы ухода за телом. Однако, в связи с тем, что нельзя однозначно отнести многие глаголы к той или иной группе, границы между группами точно не установлены. Отношение логического следования устанавливается между синсетами глаголов A и B, если из того что выполняется A, следует, что выполняется B, например, из того, что человек говорит, следует, что человек издаёт звуки.

Для установления иерархических отношений между глаголами было введено отношение тропонимии. То есть делать A означает делать B в определённой форме. Например, “Шептать – это тихо разговаривать". Отношение тропонимии – особый вид отношения следования. Отношение причины связывает два глагольных синсета, один из синсетов называется результатив, а второй каузатив. Отношение причины также может являться особым случаем отношения следования. Если A влечёт за собой B, то из B также логически следует A.

Большинство отношений WordNet связывают слова, являющиеся одной частью речи. Таким образом можно сказать, что WordNet действительно состоит из четырех подсетей, по одной для существительных, глаголов, прилагательных и наречий, с несколькими перекрестными POS-указателями.

В данной работе знания, полученные из тезаурусов, будут использоваться для разрешения лексической многозначности при переводе, что позволит избежать самостоятельного обучения системы на большом корпусе размеченных документов. Более подробно алгоритм использования тезаурусов при переводе будет описан в главе 2.

1.4 Постановка задачи

Требуется разработать мобильное приложение под ОС Android, обеспечивающее поиск в сети интернет по заданному текстовому документу релевантных ему документов. Релевантные документы могут быть в том числе представлены на языке отличном от языка заданного документа.

Поставленная задача разбивается на следующие подзадачи:

1. Разработать алгоритм составления ПОД;
2. Разработать алгоритм разрешения лексической многозначности при переводе;
3. Разработать структурно-функциональную схему системы поиска релевантных документов;
4. Реализовать мобильное приложение.

Выводы

В результате исследований предметной области получено:

1. Для осуществления поиска по заданному текстовому документу релевантных ему документов в сети интернет можно воспользоваться одним из следующих способов: разработать собственную поисковую систему; для заданного документа составить ПОД, который будет представлять собой запрос для уже существующих поисковых систем;
2. Для извлечения ключевой информации из документа используются статистические, лингвистические и гибридные методы, для повышения оценок точности и полноты работы которых нужно выполнить предварительная обработку документа;
3. Правильность перевода является одной из главных задач при поиске релевантных документов, представленных на языке отличном от языка исходного запроса.

ГЛАВА 2
АЛГОРИТМИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ЗАДАЧИ

Для составления ПОД необходимо:

1. Выполнить предварительную обработку документа: токенизация, стемминг, фильтрация стоп слов;
2. Определить для токенов важность их в контексте соответствующего документа;
3. Исходя из рассчитанных весов важности определить набор ключевых слов, который и будет являться ПОД.

Ниже будут рассмотрены популярные в IT сообществе сервисы и инструменты, позволяющие извлекать ключевую информацию из текста.

2.1 Сравнение сервисов и инструментов для извлечения ключевой информации из текста

OpenCalais представляет собой web-сервис, разработанный компанией Thomson Reuters, одной из функций которого является извлечение ключевых слов из текстов на естественном языке. Он является бесплатным и также доступен для коммерческого использования. В основе OpenCalais лежат методы обработки естественного языка, а также заранее подготовленные онтологии для различных предметных областей и машинное обучение. Первоначально над входным текстом выполняется графематическая и морфологическая разметка, затем полученные в ходе разметки словосочетания проходят идентификацию посредством обученной модели классификации именованных сущностей, между которыми осуществляется поиск семантических отношений. Полученный в результате граф сущностей и отношений между ними конвертируется в набор RDF-троек. Сервис поддерживает следующие языки: английский, французский и испанский. Ограничения на передаваемый размер файла 100кб, 50000 запросов в сутки, до 4 запросов в секунду по одному ключу.

IBM's Watson Natural Language Understanding Service предоставляет возможность определять эмоциональную окраску документа и слов в отдельности, выделять SAO, ключевые слова, а также выполнять множество других операций над текстом на естественном языке. Входными данными для данного сервиса может являться как обычный текст, html, так и url-адрес некоторого веб-сайта. Сервис предварительно очищает HTML перед анализом, удаляя большинство рекламных объявлений и другой нежелательный контент. Для извлечения ключевых слов поддерживаются следующие языки: английский, французский, немецкий, итальянский, португальский, русский, испанский, шведский. Для выделения SAO: английский и испанский. Ограничения для бесплатной версии: 1000 запросов в сутки. Размер исходного текста не должен

превышать 51 200 символов, что приблизительно эквивалентно 20 страницам текста, написанного шрифтом TimesNewRoman размером 14 пунктов.

Yahoo Content Analysis (ранее Yahoo! Term Extraction Web Service) — сервис, задействованный в работе поисковой системы Yahoo! Search. Имеет возможность обнаруживать ключевые фразы из текста на естественном языке. Подход к извлечению терминов в документации не описан. Обмен данными с пользователем осуществляется в форматах XML и JSON. Ограничение - 5000 запросов в сутки. Для коммерческого использования сервис не доступен.

Extractor набор инструментов разработчика для автоматического извлечения терминов. Предназначен для обработки естественного языка. В основе системы Extractor, согласно документации, лежит машинное обучение, генетические алгоритмы, а также статистические методы обработки естественного языка. Перед использованием систему нужно обучить на корпусе текстов, который предварительно был размечен.

Mining Cloud (ранее Text Analytics) — сервис, предназначенный для поиска информации и анализа содержания текстов, в основе которого лежат методы обработки естественного языка, а также машинное обучение. Mining Cloud позволяет пользователям встраивать текстовую аналитику и семантическую обработку в любое приложение или систему достаточно простым способом благодаря облачной инфраструктуре, с которой легко интегрироваться. Mining Cloud предоставляет следующую функциональность: извлечение темы, посредством распознавания именованных сущностей в тексте; классификация текстов через присваивание им одной или нескольких категорий в предопределенной таксономии (сервис включает несколько стандартных таксономий классификации из коробки); определение эмоциональной окраски (положительная, отрицательная, нейтральная) документа или его отдельных частей. Сервис также предлагает расширенные API-интерфейсы, такие как дополнительные тезаурусы, таксономии и т.п., оптимизированные для разных отраслей и сценариев приложений. Большинство данных API доступны на следующих языках: английском, испанском, французском, итальянском, португальском.

Stanford's Core NLP Suite предоставляет набор инструментов для анализа текста на естественном языке. Система поддерживает английский, китайский, французский, немецкий и испанский языки и включает в себя инструменты для разметки текста (разбиение текста на слова), определение базовой формы слова, части речи, извлечение именных сущностей, ключевых слов и т.д. Stanford CoreNLP предназначен для того, чтобы очень легко применить большое число инструментов лингвистического анализа к фрагменту текста, написав несколько строк кода, CoreNLP является достаточно гибким и расширяемым. Stanford CoreNLP объединяет многие инструменты Stanford’s NLP, включая частеречную

разметку, распознавание именованных сущностей, синтаксический анализатор, определение эмоциональной окраски фрагмента текста и т.д.

Natural Language Toolkit - пакет библиотек и программ, предназначенный для анализа естественного языка в приложениях, разработанных на языке Python. Он предоставляет возможность выполнять следующие операции над исходным текстом: классификация, токенизация, стемминг, тэгирование и т.д. Существует подробная документация по данному пакету, в том числе объясняющая основные концепции, встречающиеся в задачах обработки естественного языка, которые можно решить с помощью данного пакета.

Apache OpenNLP - интегрированный пакет инструментов, предназначенных для обработки текста на естественном языке и работающих на основе машинного обучения. Пакет работает на платформе Java и поддерживает наиболее распространенные задачи обработки естественного языка, такие как токенизация, сегментирование предложений, частеречная разметка, извлечение ключевых слов и т.д. Работать с данным пакетом можно посредством прикладного программного интерфейса или через командную строку. Apache OpenNLP можно использовать на условиях лицензии Apache License. Исходный код данного пакета присутствует на официальном сайте проекта.

В качестве сервиса/инструмента для извлечения ключевой информации из текста в данной работе будет использоваться IBM's Watson Natural Language Understanding Service. При выборе учитывались следующие параметры:

- Простота интеграции и отсутствие необходимости в поднятии собственного сервера;
- В качестве входных данных можно передавать, как обычный текст, так и url-адрес веб-страницы, в последнем случае сервис на этапе предварительной обработки очистит веб-страницу от рекламы и другого нежелательного контента;
- При извлечении ключевых слов поддерживаются языки: английский, русский, французский, немецкий, итальянский, португальский, шведский, испанский;
- Наличие бесплатной версии API.

Перед началом взаимодействия с IBM's Watson Natural Language Understanding Service нужно зарегистрироваться в среде IBM BLUEMIX. По окончанию регистрации будут выданы логин и пароль, которые в последствии используются при отправке запросов.

В качестве примера использования данного сервиса будет выполнено извлечение ключевых слов из статьи на английском языке, посвященной лечению рака. Входными данными для сервиса является url данной статьи: https://www.cancer.gov/about-cancer/treatment отправка которого будет осуществляться на адрес: https://gateway.watsonplatform.net/natural-language-understanding/api/v1/analyze?version=2017-02-27. Список необходимых

операций, которые надо выполнить над исходным текстом, передаётся через параметр “features”: features= keywords. Количество извлекаемых ключевых слов ограничивается через параметр: keywords.limit=6. Ответ с сервера будет иметь следующий вид (см. рисунок 2.1).

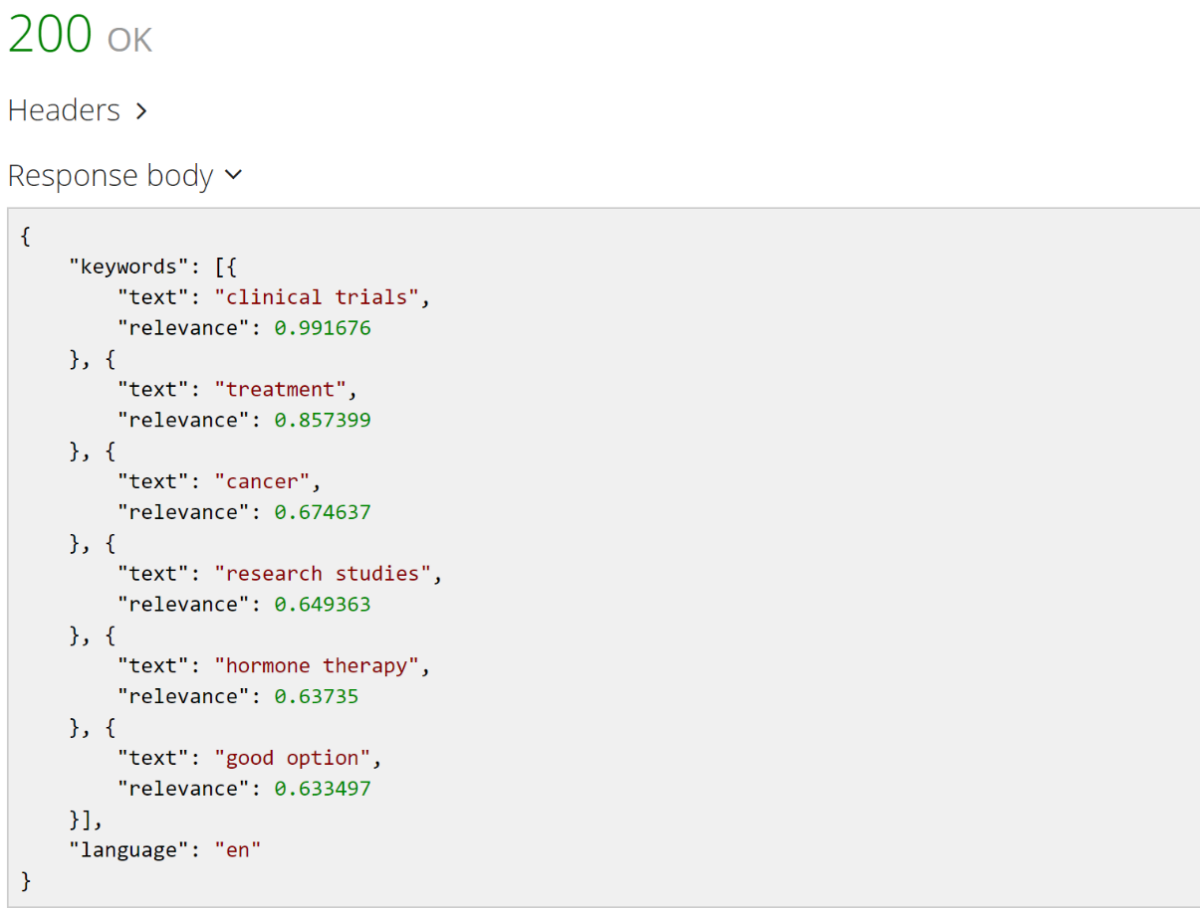

Рисунок 2.1 – Извлечённые из статьи ключевые слова

2.2 Разрешение лексической многозначности слов при переводе

Языковое выражение является неоднозначным/многозначным, если оно одновременно имеет нескольких различных смыслов. Многозначность подразделяется на следующие типы: лексическую, синтаксическую и речевую. В рамках данной работы будет рассматриваться разрешение именно лексической многозначности(WSD). Например, слово “ключ” может употребляться в одном из следующих значений: ключ как инструмент для открывания и ключ как источник воды.

Для разрешения многозначности используются система словарных знаний, включающая в себя множества значений слов и корпус текстов для разрешения. В качестве данных, на которые опирается процесс WSD, могут быть взяты

корпусы текстов, словари, тезаурусы, глоссарии, онтологии и т. д. При оценке качества WSD используются два параметра: точность и полнота разрешения.

Методы разрешения лексической многозначности подразделяют на следующие категории: методы, использующие внешние источники информации (тезаурусы, лексические базы данных и т.п.), методы, в основе которых лежит машинное обучение с учителем, и методы, являющиеся комбинацией 1-ых и 2-ых.

Ниже будет рассмотрено несколько методов разрешения лексической многозначности, которые можно разбить на следующие группы:

- методы, использующие нейронные сети;
- методы, использующие ансамбль байесовских классификаторов;
- методы, основанные на контекстной кластеризация;
- методы, использующие внешние источники информации.

2.2.1 WSD на основе нейронных сетей

Нейронная сеть на вход принимает слово, значение которого нужно определить. Для каждого значения слова есть свой узел. С учётом разрешения лексической многозначности для тренировочных целевых слов, в ходе обучения настройка весов связующих узлов соединений выполняется так, чтобы по окончании обучения узел, имеющий наибольшую активность, являлся узлом истинного значения целевого слова [8]. Во время обучения сети используется метод обратного распространения. Веса соединений настраиваются посредством рекуррентных алгоритмов и могут быть как положительными, так и отрицательными. Сеть также может включать в себя скрытые слои, связи между узлами которых могут быть как прямыми, так и обратными.

Узел, соответствующий целевому слову, имеет соединение со смысловыми узлами посредством активирующих связей. Смысловые узлы соответствуют всем возможным значениям слова, встречающимся в различных словарных статьях. В то же время каждый смысловой узел соединяется посредством активирующих связей с узлами, связанными со словами в словарной статье, в которой целевое слово употребляется в данном значении. Процесс соединения повторяется многократно. Таким образом создаётся большая сеть взаимосвязанных узлов, которая в лучшем случае будет содержать весь словарь.

Во время запуска сети сначала происходит активация узлов для входного слова. После этого смысловые узлы получают активирующий сигнал от входных узлов, с которыми они имеют соединение. В течении нескольких циклов сигналы будут распространены по всей сети. Во время выполнения каждого из циклов до узла слова и узлов его значений будут доходить обратные сигналы, поступающие от соединённых вместе с ними узлов. Узлы различных значений одного и того

же слова отправляют взаимно подавляющие сигналы. Увеличение активности узлов слов и соответствующих для них узлов истинных значений происходит посредством взаимодействия сигналов подавления и сигналов обратной связи. В то же время уменьшается активность узлов, соответствующих неправильным значениям целевого слова. По выполнению некоторого количества циклов сеть будет стабилизирована и установится состояние, когда активированы только узлы значений, для которых связи с узлами слов являются наиболее активированными.

2.2.2 Использование ансамбля байесовских классификаторов для разрешения многозначности

При использовании данных WSD методов на выбор значения влияет частота взаимного попадания слов в окно заранее установленного размера в текстах корпуса. Для разрешения лексической многозначности к размеченному корпусу применяют методы машинного обучения с учителем и различные статистические методы, в которых словам корпуса, для которых установлено значение, ставится в соответствие некоторый набор языковых свойств [9, 10].

Вероятностный классификатор, в основе которого лежит применение теоремы Байеса, называется наивным байесовским классификатором. Для разрешения лексической многозначности происходит объединение ряда наивных байесовских классификаторов в ансамбль, выбор значения происходит посредством голосования простым большинством голосов. Также вводится понятие контекста, которому принадлежит многозначное слово. Контекст представляет собой функцию переменных $(F_1, F_2, \dots, F_n)$, где все переменные являются бинарными. Классификационная переменная S будет обозначать некоторое значение многозначного слова. Переменная, ставящаяся в соответствие некоторому слову из контекста, будет принимать значение “true”, в случае, если расстояние до данного слова не будет превышать некоторого заранее установленного количества слов, находящихся справа или слева от целевого слова. Вероятность взаимной встречаемости набора переменных контекста с определённым значением слова вычисляется по следующей формуле:

$$p(F_1, F_2, .., F_n, S) = p(S) \prod_{i=1}^{n} p(F_i \mid S) \qquad (2.5)$$

Для оценки параметров необходима информация о частоте событий, характеризуемых взаимозависимыми переменными $(F_i,\ S)$. Частота событий равна числу предложений, в которых слово, представляемое как F_i, принадлежит некоторому контексту многозначного слова, имеющего для данного контекста

значение S. Если появляются нулевые параметры, то для сглаживания по умолчанию им присваивается крайне маленькое значение. По окончанию оценивания всех параметров модель можно считать обученной и использовать в качестве классификатора.

В основе контекста лежит модель "мешок слов". Согласно данной модели выполняется предварительная обработка текста, в ходе которой все слова будут переведены в нижний регистр, после чего над ними будет выполнена лемматизация, так же удаляются все знаки препинания из текста. Контексты подразделяются на левое и правое окно. Левое окно содержит слова, которые находятся слева от неоднозначного слова. Правое окно содержит слова, которые, соответственно, находятся справа от неоднозначного слова. Размер окон контекстов может быть одним из следующих 9: 0, 1, 2, 3, 4, 5, 10, 25, 50 слов. Таким образов всего возможно 81 сочетание левого и правого размера окон, для каждого из которых будет обучен отдельный наивный байесовский классификатор на первом шаге в ансамблевом подходе. Наивный байесовский классификатор (l, r) содержит l слов, которые находятся слева от неоднозначного слова и, соответственно, r слов, которые находятся справа. Классификатор (0, 0,) будет представлять собой единственное исключение, так как не содержит слов ни слева, ни справа. Из-за пустого контекста данному классификатору будет присваиваться априорная вероятность слова, которая равна вероятности принятия словом наиболее употребляемого значения. На следующем шаге построения ансамбля выбираются классификаторы, которые станут его членами. Все классификаторы разбиваются на категории в зависимости от диапазона размеров окна контекста. Выделяются узкий, средний и широкий диапазоны. К узкому относятся окна шириной в 0-2 слова, к среднему окна шириной 3-5 слов, а к широкому, соответственно, 10, 25, 50 слов. Таким образом получается 9 возможных комбинаций, по 3 на левое и по 3 на правое окно. В данном случае наивный байесовский классификатор (3, 10) будет принадлежать к категории (средний, широкий), так как его окно слева имеет размерность 3 слова, а окно справа 10 слов. Для включения в ансамбли выбираются классификаторы, которые являются самыми точными в своей категории. После этого каждый из 9 классификаторов, учитывая контекст, будет определять наиболее вероятное значение слова. Ансамбль разрешает многозначность путем выбора в качестве значения целевого слова значение, получившее максимальное количество голосов.

2.2.3 Контекстная кластеризация

Каждое вхождение многозначного слова в корпус характеризуется некоторым контекстным вектором. Для контекстных векторов выполняется

алгоритм кластеризации, в основе которого лежит дистрибутивная гипотеза, что если слова встречаются в схожих контекстах, то они являются близкими по смыслу. Таким образом разные значения слова будут относиться к разным кластерам. Методы кластеризации обычно используются для разрешения лексической многозначности слов без учителя при небольшом количестве обучающих данных.

Для разрешения лексической многозначности используются контекстные векторы. Контекстный вектор представляют собой средний вектор по векторам свойств всех слов, принадлежащих контексту. Вектор свойств описывает взаимное употребление данного слова вместе с остальными словами. Вычисление векторов свойств выполняется на этапе обучения системы.

Сначала выполняется построение по данным обучающего корпуса матрицы взаимного употребления слов. Строка матрицы является вектором свойств, тем самым описывает встречаемость данного слова с другими. По окончанию построения матрицы происходит разбиение тестовых данных, согласно которому целевые слова группируются вместе с примерами их употребления. Затем осуществляется переход от тестовых данных, содержащих анализируемое слово, к набору контекстных векторов, соответствующих некоторому употреблению исходного слова.

Получающиеся в ходе кластеризации контекстных векторов кластеры состоят из употреблений близких по значению фраз. Разрешение лексической многозначности происходит путём определения к какому кластеру относится целевое слово, учитывая, что каждому значению многозначного слова ставится в соответствие отдельный кластер. Полученные в результате обучения на небольшом корпусе текстов векторы свойств могут иметь маленькую размерность, а это затруднит процесс описания закономерности взаимного употребления слов. Поэтому в данном случае векторы свойств слов увеличиваются за счёт содержательных слов соответствующих им словарных определений.

2.2.4 WSD на основе тезаурусных знаний

Одним из способов использования тезаурусных знаний является расчёт семантической близости между контекстом вхождения многозначного слова и всеми синсетами, каждый из которых соответствует одному из значений данного слова. Такой способ можно реализовать посредством сравнения близости путей между синсетами слов, принадлежащих контексту, и синсетами слова, значение которого в данном контексте хотим определить.

В качестве примера будет рассмотрен метод Леска [11], который основан на поиске значения слова в списке словарных определений с учетом контекста, в

котором используется данное слово. Основным критерием при выборе значения является следующее правило: заложенный в определении смысл должен частично совпадать со смыслом значений соседних слов в контексте.

Метод леска можно разбить на следующие шаги:

1. Для исходного слова выделяется контекст, размер которого не более 5 ближайших по расположению слов;
2. Для каждого слова из контекста осуществляется поиск всех его определений в словаре. V - множество слов, содержащихся в определениях;
3. Сопоставляется каждое определение исходного слова с V. В случае если какое-либо из слов, принадлежащих V, присутствует в определении, то этому определению дается балл;
4. Наиболее вероятным значением является то, определение которого набрало наибольшее количество баллов.

При определении значения слова актуального данному контексту в конструкции более длинной, чем несколько слов, так же можно использовать упрощенный алгоритм Леска. В котором пересечение осуществляется между описаниями значений слов и контекстами данных слов в тексте. Кроме толкований словаря для улучшения точности можно дополнительно использовать размеченные корпуса, а также примеры использования различных значений данного слова.

В качестве одного из примеров использования метода Леска для разрешения лексической многозначности на основе тезауруса WordNet нужно определить значение всех слов, входящих в словосочетание “pine cones”. Для каждого из слов имеем следующие таблицы (см. таблица 2.1 – 2.2), в столбцах которых указаны соответственно: слово, номер значения, количество пересечений со значениями других слов из контекста, часть определения.

Таблица 2.1 — Таблица значений “pine”

Слово	№	Пересечений	Определение
pine	1	3	kinds of evergreen tree with needle-shaped leaves
cone	2	0	waste away through sorrow or illness

Таблица 2.2 — Таблица значений “cone”

Слово	№	Пересечений	Определение
cone	1	0	solid body which narrows to a point
cone	2	1	something of this shape whether solid or hollow
cone	3	2	fruit of certain evergreen trees

Максимальное пересечение достигаются между первым определением слова “pine” и третьем определением слова “cone”, следовательно, эти значения являются наиболее подходящими согласно методу Леска. Недостатком этого метола является, то что при разрешении многозначности очередного слова не учитываются уже найденные значения других слов из контекста, таким образом алгоритм выполняется для каждого слова отдельно.

В данной работе для разрешения лексической многозначности при переводе будет использован метод Леска, основанный на использовании тезаурусных знаний, что позволит избежать самостоятельного обучения системы на большом корпусе размеченных документов. Основные этапы применения метода следующие: для исходного ключевого слова выделяется контекст, для всех слов контекста из многоязычного тезауруса извлекаются соответствующие им синсеты, после чего рассчитывается вероятность употребления ключевого слова в каждом из его значений в данном контексте. В качестве многоязычного тезауруса будет использоваться BabelNet.

BabelNet - это многоязычный энциклопедический словарь материалы которого доступны на 271 языке с лексикографическим и энциклопедическим охватом терминов, а также семантическая сеть, которая связывает понятия и именованные сущности большой сетью семантических отношений, состоящей из около 15 миллионов синсетов. Каждый синсет представляет собой определенное значение и содержит все синонимы, которые выражают это значение на разных языках.

На данный момент BabelNet получается из автоматической интеграции:

- WordNet (версия 3.0);
- Open Multilingual WordNet (январь 2017);
- OmegaWiki - большой многоязычный словарь (январь 2017);
- Wikipedia - крупнейшая многоязычная веб-энциклопедия (январь 2017);
- Wiktionary (февраль 2017);
- Wikidata (январь 2017);
- Wikiquote - многоязычный сборник цитат и творческих работ (март 2015);
- VerbNet (версия 3.2);
- Microsoft Terminology (июль 2015);
- GeoNames - свободная географическая база данных, содержащая более восьми миллионов названий городов (апрель 2015);
- WoNeF - французский перевод WordNet (февраль 2017);
- ItalWordNet - лексико-семантическая база данных для итальянского языка (февраль 2017);
- ImageNet - база данных изображений, организованная в соответствии с иерархией WordNet (2011);

- FrameNet (версия 1.6);
- WN-Map - сопоставления между версиями WordNet (2007);
- Korean WordNet (январь 2017);
- GAWN WordNet - база данных, состоящая из ирландских слов и семантических отношений между ними (январе 2017).

Использовать данный тезаурус можно бесплатно под лицензией CC BY-NC-ND 4.0. Ограничение на число запросов - 50000 в сутки.

В качестве примера использования BabelNet будет выполнено разрешение лексической многозначности при переводе первых 3 ключевых слов, извлечённых с помощью IBM's Watson Natural Language Understanding Service в предыдущей главе из статьи, посвящённой лечению рака. Перевод будет осуществляться на русский и французский языки. Множество синсетов, извлечённых из BabelNet, для ключевых слов “clinical trials”, “treatment”, “cancer” приведены ниже (см. Таблица 2.3-2.5).

Таблица 2.3 – Синсеты для “clinical trials”

Синсет и соответствующие ему значения	Русский перевод	Французский перевод
clinical trial, clinical test, Clinical trials, Clinical research trial, Clinical researcher, Clinical studies, Clinical study, Comparator Sourcing, Controlled clinical trial, Controlled trials, Device Clinical Trials, Drug studies, Drug tester, Drug trial, Human testing, Non-controlled studies, Novoclinica, Online clinical trial, Online clinical trials, Pharmaceutical testing, Pharmaceuticals testing, Placebo group, Study population, Uncontrolled trial. A rigorously controlled test of a new drug or a new invasive medical device on human subjects; in the United States it is conducted under the direction of the FDA before being made available for general clinical use. Trials are prospective biomedical or behavioral research studies on human subjects that are designed to answer specific questions about biomedical or behavioral interventions, generating safety and efficacy data.	Клинические исследования, Клинические испытания	Essai clinique, étude clinique

Таблица 2.4 – Синсеты для “treatment”

Синсет и соответствующие ему значения	Русский перевод	Французский перевод
treatment, intervention Provided to improve a situation (especially medical procedures or applications that are intended to relieve disease or injury). Medical care for an disease or injury. A treatment or cure is applied after a medical problem has already started.	лечение	traitement
treatment, handling The management of someone or something.	обращение	traitement
A manner of dealing with something artistically	трактовка	traitement
discourse, treatment, discussion, speech An extended communication dealing with some particular topic.	Дискурс, доклад, лекция	discours, discours politique, discours public

Таблица 2.5 – Синсеты для “cancer”

Синсет и соответствующие ему значения	Русский перевод	Французский перевод
cancer, malignant neoplastic disease, malignant neoplasm, malignant tumor, primary cancer Any malignant growth or tumor caused by abnormal and uncontrolled cell division; it may spread to other parts of the body through the lymphatic system or the blood stream. Group of diseases defined by unregulated cell growth and proliferation. A disease in which the cells of a tissue undergo uncontrolled (and often rapid) proliferation. Cancer is a class of diseases in which a group of cells display uncontrolled growth. Disease of uncontrolled cellular proliferation.	рак, злокачественная опухоль	cancer, tumeur maligne
Cancer, Cancer constalation, Cancer constellation, Carcinos, Carcinus A small zodiacal constellation in the northern hemisphere; between Leo and Gemini.	Рак	Cancer

A constellation of the zodiac supposedly shaped like a crab. Cancer is one of the twelve constellations of the zodiac. Cancer is the fourth astrological sign, which is associated with the constellation Cancer.		
Cancer are a British death/thrash metal band formed in Ironbridge, Telford, Shropshire in 1988.	Cancer	Cancer
Cancer is the first full-length studio album by Australian hardcore/metalcore band Confession, released on 10 September 2009, through Resist Records.	Cancer	Cancer

Из приведённых выше таблиц следует, что “clinical trials” переводится однозначным образом как “клинические исследования”/“essai clinique” на русский/французский язык соответственно.

Для разрешения лексической многозначности при переводе “treatment” и “cancer” нужно сначала выделить контекст их употребления. Данные слова встречаются в предложении “The types of treatment that you have will depend on the type of cancer you have and how advanced it is.”. SAO для данного предложения будет выделено с помощью IBM's Watson Natural Language Understanding Service с параметром features=semantic_roles (см. Рисунок 2.2).

Рисунок 2.2 – SAO

Таким образом контекстом для “treatment” и “cancer” будет являться “The types of treatment depend on the type of cancer you have”. Согласно методу разрешения лексической многозначности Леска для данного контекста у переводов “лечение”/“traitement” и “рак”/“cancer” максимальное количество пересечений(по 2), следовательно, они и будут выбраны. Если у нескольких значений многозначного слова было одинаковое количество пересечений, то в данном случае выбиралось бы наиболее часто употребляемое значение.

Выводы

1. Исходя из относительно высокой оценки точности и полноты извлечения ключевых слов, поддержке английского, русского, французского, немецкого, итальянского, португальского, шведского и испанского языков, возможности приёма в качестве входных данных url веб-страницы, что особенно актуально для мобильных клиентов, а также наличии бесплатной версии API, при составлении ПОД будет использоваться IBM's Watson Natural Language Understanding Service;
2. Для выполнения перевода многозначного ключевого слова в указанном контексте будет использоваться алгоритм Леска, для которого в качестве внешнего источника знаний выбран многоязычный энциклопедический словарь BabelNet, интегрирующий в себе WordNet, Wikipedia, а также множество других ресурсов и покрывающий 271 язык.

ГЛАВА 3
МОБИЛЬНЫЙ КЛИЕНТ ДЛЯ ПОИСКА РЕЛЕВАНТНЫХ ДОКУМЕНТОВ

Мобильное приложение, разрабатываемое под операционную систему Android и предоставляющее функционал поиска по заданному текстовому документу релевантных ему документов в сети интернет, будет обладать удобным и приятным интерфейсом, простым в использовании, поддерживать весь спектр существующих устройств с версией API 14+ (Android 4.0). Интерфейс для работы с приложение будет разработан в соответствии с концепцией material design.

Для извлечения ключевой информации из поступающего на вход текста/веб-страницы будет использоваться IBM’s Watson Natural Language Understanding Service, для разрешения лексической многозначности - многоязычный энциклопедический словарь BabelNet.

Язык поступающего на вход текстового документа/веб-страницы - английский. Язык обнаруженных релевантных документов может быть одним из 271 перечисленных здесь http://babelnet.org/stats#LanguagesandCoverage.

3.1 Структурно-функциональная схема системы поиска релевантных документов

Входными данными для разрабатываемой системы является текстовый документ либо адрес веб-страницы. Язык исходного текста/веб-страницы принадлежит множеству $L = W \cap B,$ где W-множество языков, для которых IBM's Watson Natural Language Understanding Service умеет извлекать ключевые слова, B - множество языков, поддерживаемых многоязычным энциклопедическим словарём BabelNet.

Выходные данные - набор запросов для поисковой системы Google. Языки поисковых запросов принадлежат множеству B.

Поиск релевантных документов в многоязычной информационной среде будет осуществляться по схеме, представленной на рисунке 2.3.

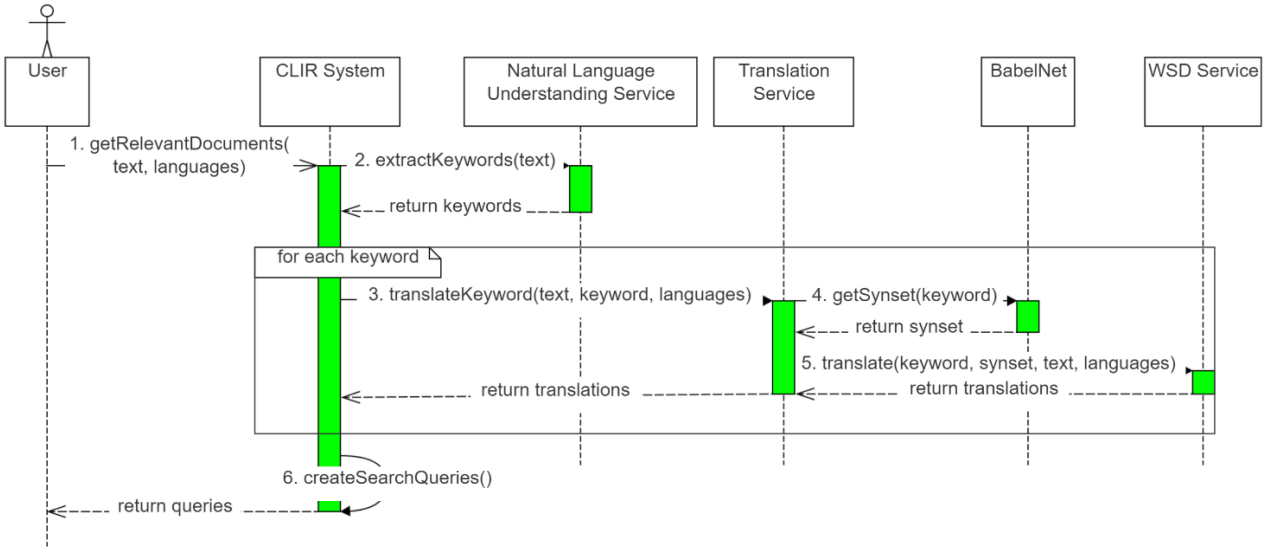

Рисунок 2.3 – Структурно-функциональная схема поиска релевантных документов

1. Запрос на поиск релевантных документов;
2. Составление ПОД с использованием IBM's Watson Natural Language Understanding Service;
3. Запрос на перевод ключевого слова, входящего в ПОД;
4. Извлечение из многоязычной энциклопедии BabelNet всех синсетов, содержащих данное ключевое слово;
5. Разрешение лексической многозначности при переводе ключевого слова согласно алгоритму Леска;
6. Формирование поисковых запросов для Google.

3.2 Проектирование архитектуры приложения

При разработке приложения используется объектно-ориентированный подход. Архитектура приложения будет построена по принципу Clean Architecture.

Clean Achitecture — принцип разработки приложений, предложенный Uncle Bob'ом. Код, спроектированный с учётом данной архитектуры, легче тестировать и переиспользовать.

Преимуществами Clean Achitecture являются:

- Независимость от внешних сервисов, с которыми взаимодействует приложение;
- Независимость от фреймворков;
- Независимость от UI;
- Независимость от Баз Данных;

- Простота написания тестов.

Суть Clean Architecture заключается в разделении логики приложения на несколько составляющих слоёв: слой бизнес-логики, слой представления и слой данных. При этом чтобы обеспечить максимальную независимость между слоями, на каждом из них используется своя модель данных, которая конвертируется при взаимодействии между слоями. Для взаимодействия между слоями выделяются отдельные интерфейсы.

Схема данных слоев представлена на рисунке 3.1.

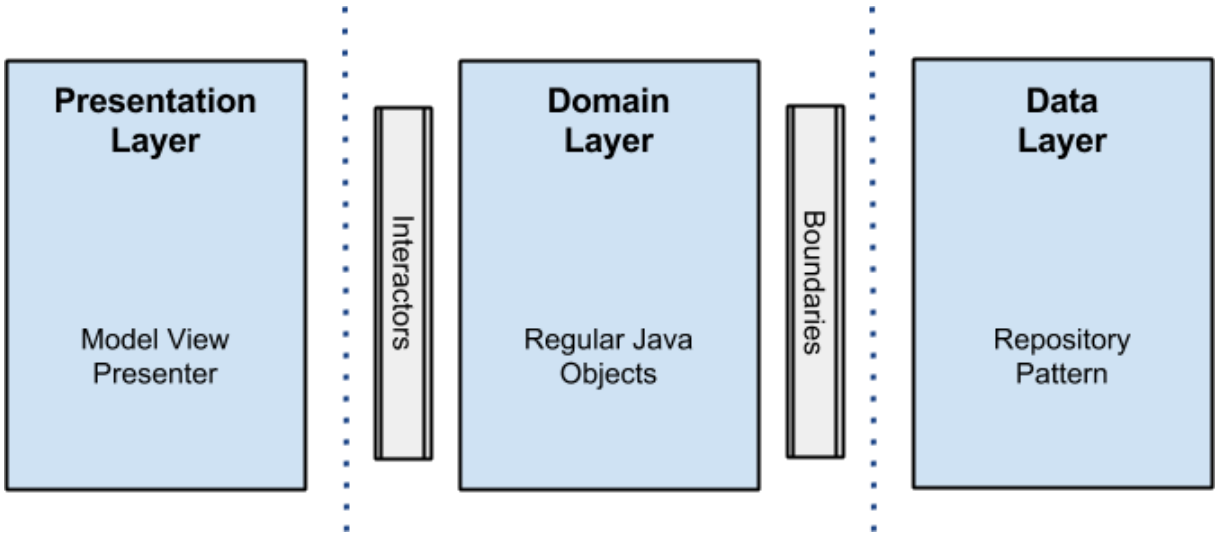

Рисунок 3.1 – схема Clean Architecture

Слой представления предназначен в первую очередь для взаимодействия с пользователем, так же он отвечает за логику отображения данных на экране и за другие процессы, связанные с UI. Этот слой не должен содержать логику приложения, не связанную с UI. Именно слой представления привязывается к экранам и помогает организовать взаимодействие со слоем бизнес-логики и работу с данными. Данный слой может быть реализован с использованием любого предпочитаемого паттерна, к примеру, MVC, MVP, MVVM и других.

При разработке приложения слой представления будет организован согласно паттерну MVP. Он позволит разделить экран на UI-часть (View), на логику работы с UI (Presenter) и объекты для взаимодействия с UI (Model).

В MVP Presenter управляет только одной View и взаимодействует с ней через специальный интерфейс. View управляется только с помощью Presenter и не отслеживает изменения Model. Presenter получает все данные из слоя данных, обрабатывает их в соответствии с требуемой логикой и управляет View.

Слой бизнес-логики содержит всю бизнес-логика приложения. Этот слой является неким объединением слоев сценариев взаимодействия и бизнес-логики.

Именно к нему обращается слой представления для выполнения запросов и получения данных. Слой бизнес-логики будет реализован в виде Java-модуля, который не содержит никаких зависимостей от Android-классов. Преимуществом данного подхода является то, что для реализации бизнес-логики нужны только классы моделей и стандартные средства языка Java. Более того, такой подход позволит легко тестировать слой бизнес-логики с помощью обычных тестов на JUnit, что очень удобно.

Слой данных отвечает в первую очередь за получение данных из различных источников и их кэширование. Он реализуется согласно паттерну Repository, общую схему представлена на рисунке 3.2.

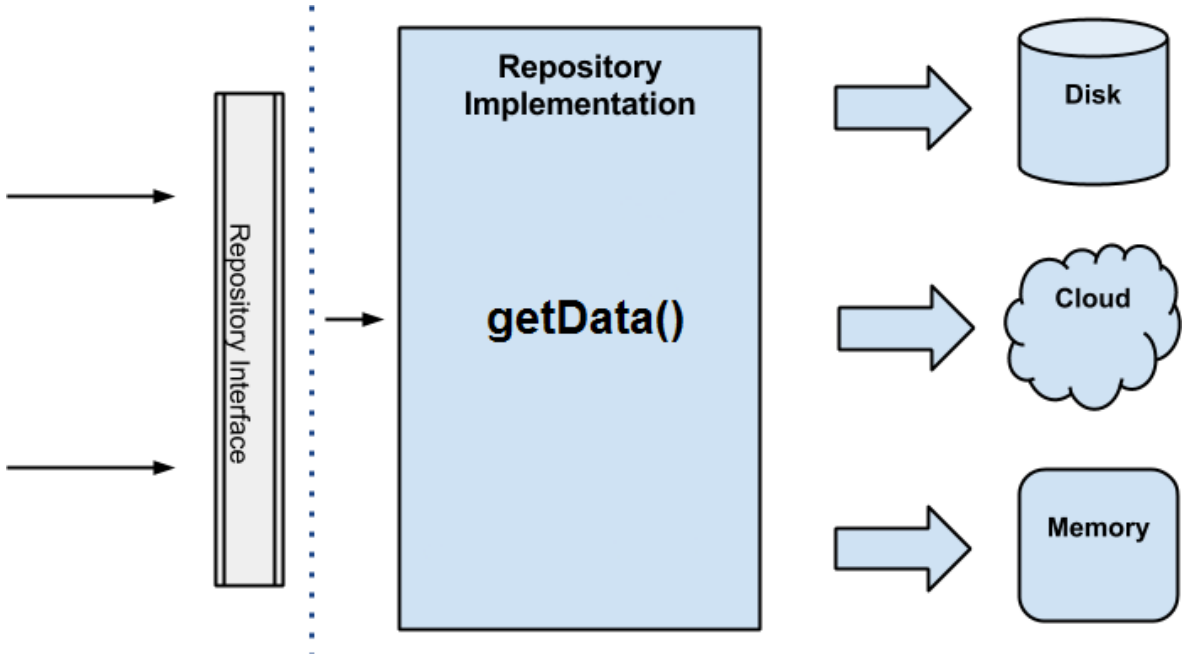

Рисунок 3.2 – Слой данных

Существует несколько плюсов от использования такого подхода. Во-первых, другие слои, которые запрашивают данные, не знают о том, откуда эти данные приходят. Более того, им не нужно этого знать, так как это усложняет логику работы и модуль берет на себя лишнюю ответственность. Во-вторых, слой данных в таком случае выступает единственным источником информации.

3.2 Методика применения разработанного приложения

Разработанная программа обладает интуитивно понятным интерфейсом.

Для поиска в сети интернет документов релевантных данному тексту/веб-странице нужно выполнить действия, описанные ниже. Сначала необходимо

запустить приложение. После заставки отобразится главный экран (см. Рисунок 3.3).

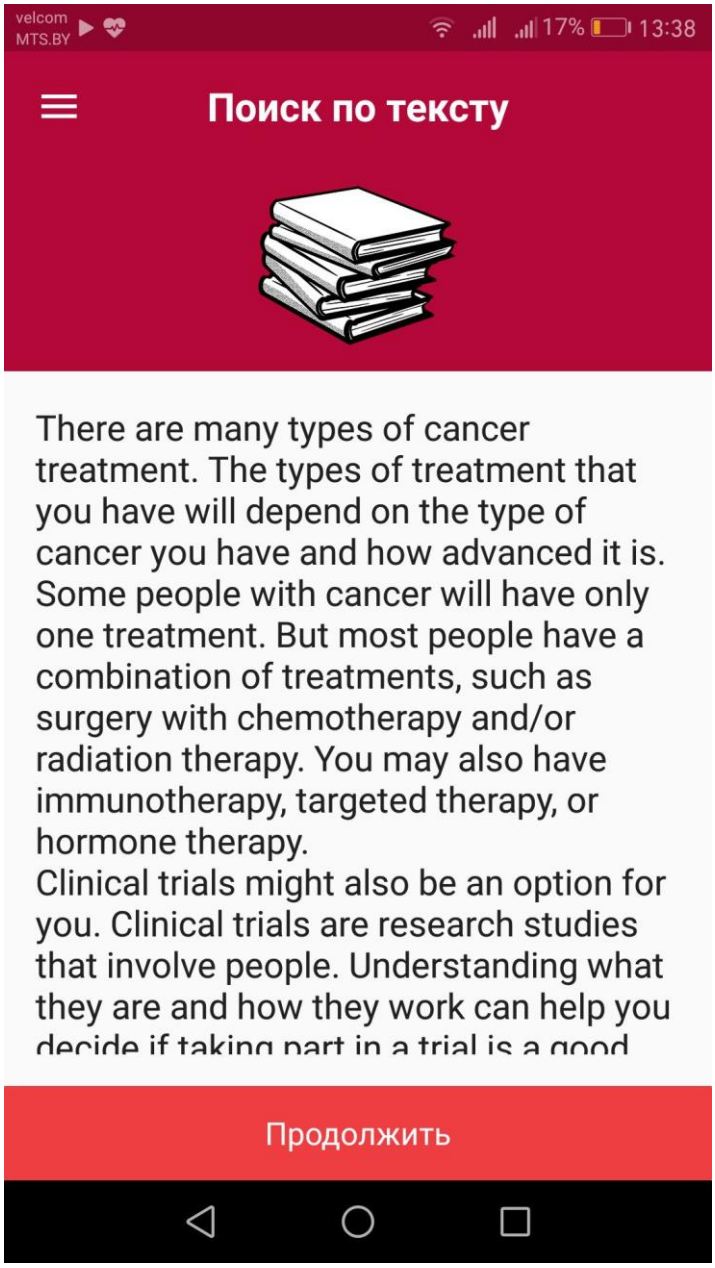

Рисунок 3.3 – Главный экран приложения

На главном экране в появившемся поле ввода необходимо ввести текст, для которого будет осуществляться поиск релевантной информации в интернете. Так же в качестве исходных данных можно использовать url веб-страницы. Для этого нужно перейти на экран “Поиск по url”, доступный через навигационное меню. Для отображения навигационного меню можно выполнить свайп слева-направо по экрану, либо нажать на кнопку “Меню”.

По нажатию кнопки “Продолжить” появится экран выбора языков, на которых будет извлекаться актуальная информация из сети интернет (см. Рисунок 3.4).

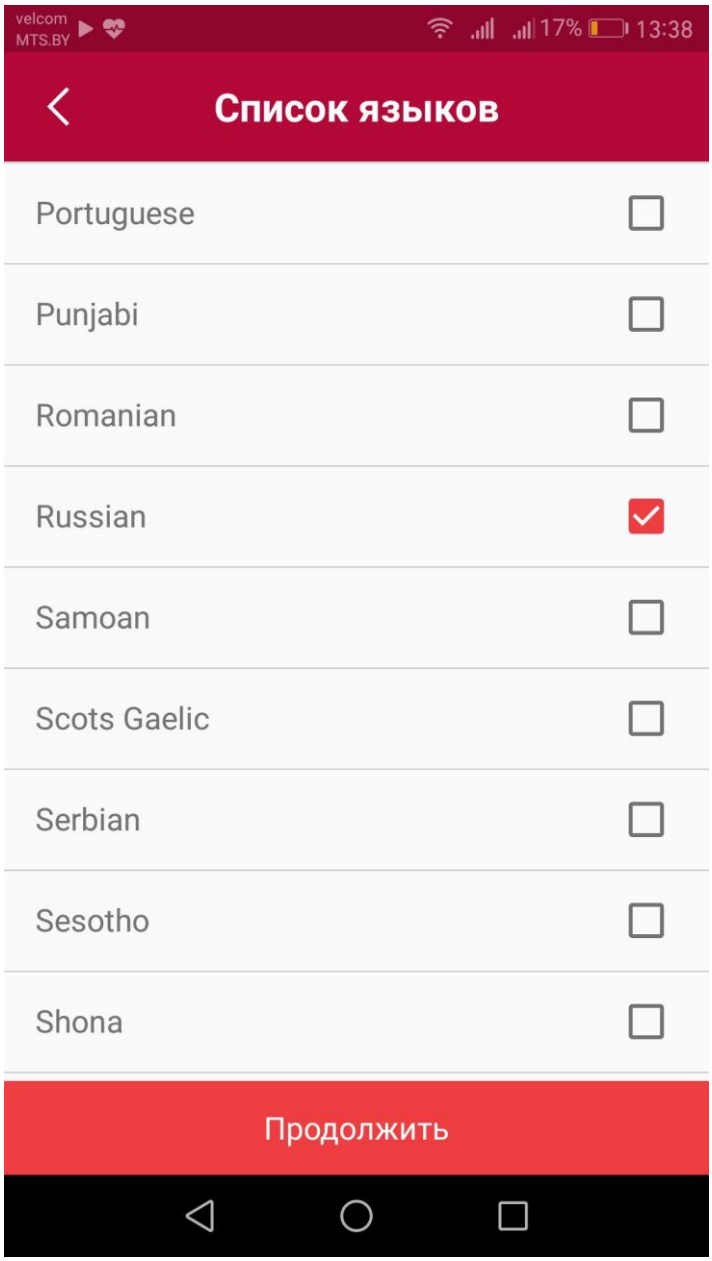

Рисунок 3.4 – Экран выбора актуальных пользователю языков

На данном экране выбора отображены все поддерживаемые приложением для поиска информации языки.

По нажатию кнопки “Продолжить” после обработки входных данных будет выдан список сформированных приложением запросов для поиска релевантных данному документов на выбранных пользователем языках. В дальнейшем список языков можно будет поменять, не вводя текст заново.

Экран сформированных запросов и экраны результатов поиска в поисковой системе Google приведены ниже (см. Рисунок 3.5).

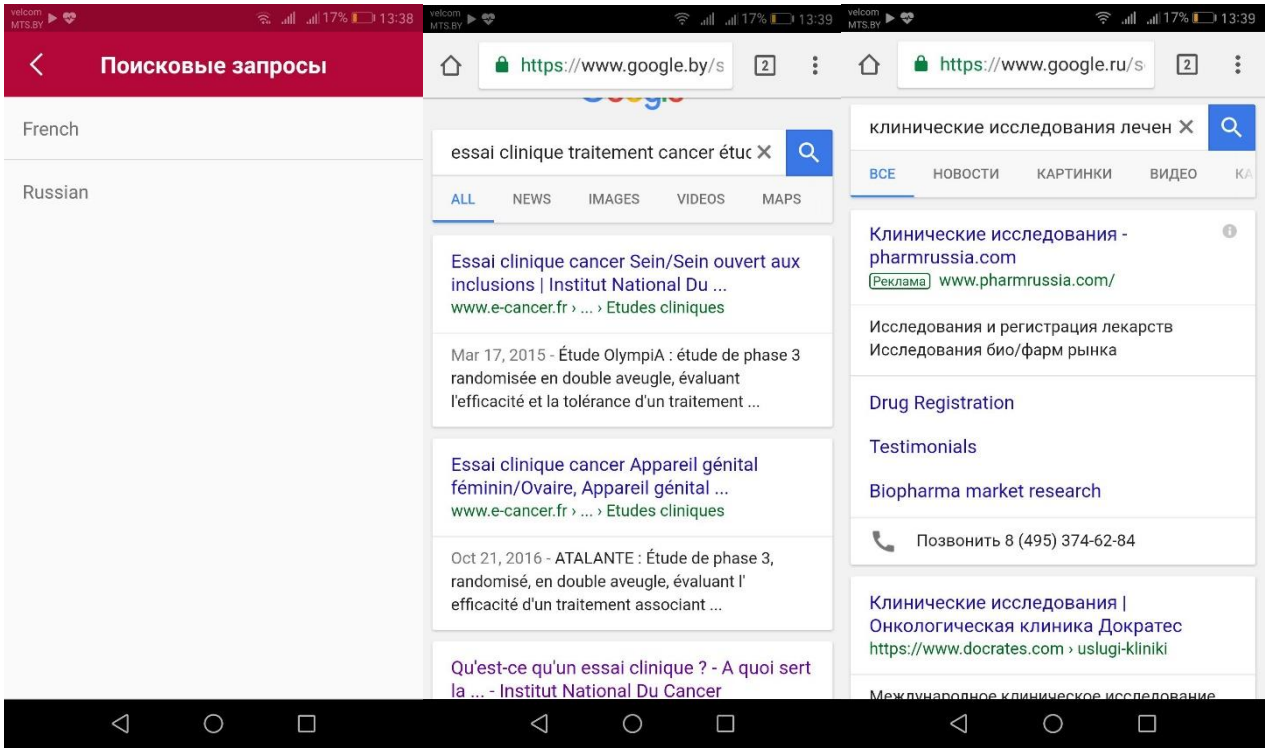

Рисунок 3.5 – Результаты поиска релевантных документов

Так же через навигационное меню можно просмотреть историю поиска и перейти на экран настроек приложения (см. Рисунок 3.6). На экране настроек можно поменять цветовую гамму и установить языки, которые будут выбраны при поиске информации по умолчанию. В истории вместе с входными данными так же хранятся уже сформированные запросы для Google. При необходимости можно очистить историю поиска.

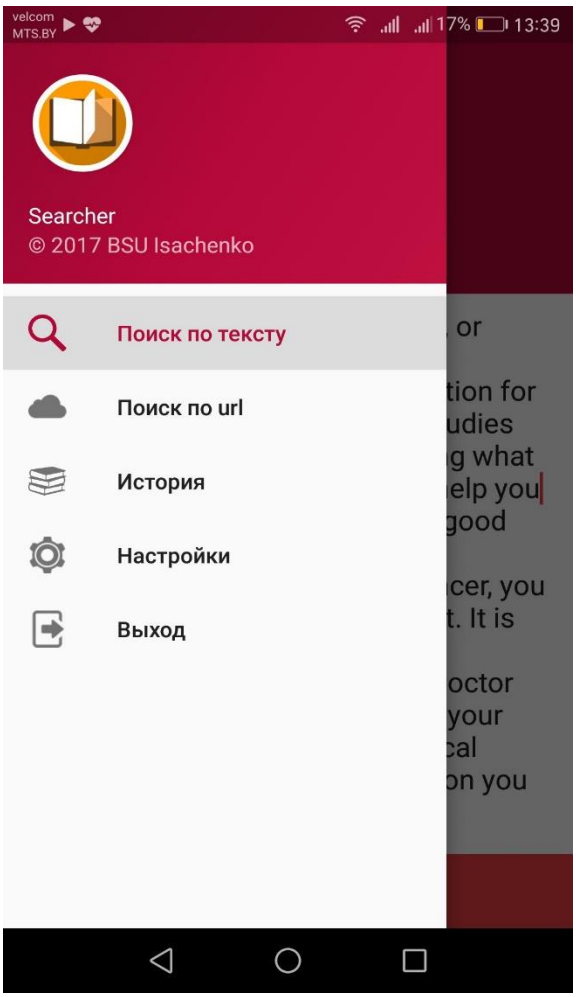

Рисунок 3.6 – Навигационное меню приложения

По клику на пункт “История” откроется экран истории поиска. По клику на экран “Настройки” откроется экран настроек. Для выхода из приложения можно использовать кнопку “back” или пункт “Выйти” в навигационном меню.

Выводы

1. Построена структурно-функциональная схема поиска для заданного документа/веб-страницы релевантных документов, представленных в том числе на языке отличном от языка входных данных;
2. Спроектирована архитектура приложения согласно подходу “Clean architecture”;
3. Описана методика применения разработанного приложения.

ЗАКЛЮЧЕНИЕ

В рамках дипломной работы «*Cross-language функциональность автоматического поиска в сети Internet релевантных документов*» получены следующие результаты.

В первой главе выполнен анализ подходов к реализации многоязычного поиска. Сформулированы цели и задачи для поставленной проблемы. Рассмотрены методы извлечения ключевой информации из текста, основанные на семантических данных о словах и численных характеристиках встречаемости слов в тексте. Показана роль перевода при CLIR, а также рассмотрены преимущества каждого из подходов его реализации.

Во второй главе исследованы все самые известные на текущий момент алгоритмы, которые применяются для разрешения лексической многозначности при переводе слов, описаны их преимущества и недостатки. Данные алгоритмы можно подразделить на 3 класса: алгоритмы, использующие внешние источники информации, алгоритмы, базирующиеся на машинном обучении, работающие на размеченных корпусах текстов, а также алгоритмы, представляющие собой комбинацию 1-ых и 2-ых. Так же выполнен анализ сервисов, предоставляющих возможность извлечения ключевой информации из текста.

В третьей главе построена структурно-функциональная схема поиска для заданного документа/веб-страницы релевантных документов, представленных в том числе на языке отличном от языка входных данных. Спроектирована архитектура приложения согласно подходу “Clean architecture”. Также приведена методика применения, разработанного под ОС Android приложения.

CLIR является очень актуальной задачей, однако точность многоязычного поиска на данный момент невысока, одной из причин является сложность разрешения лексической многозначности слов при переводе. Возможно уже в ближайшем будущем, данный тип поиска позволит устранить лингвистическое несоответствие между предоставляемыми запросами и документами, которые извлекаются из информационной сети, тем самым убрав языковой барьер между информацией, представленной на разных языках.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Porter, M.F. An algorithm for suffix stripping. / M.F. Porter – Cambridge, 1997. – 6 p.
2. Turney, P.D. Learning algorithms for keyphrase extraction. Information Retrieval / P.D. Turney. - Ottawa, Ontario, Canada, 2000. – 477 p.
3. Matsuo, Y. Keyword Extraction from a Single Document using Word Co-occurrence Statistical Information. / Y. Matsuo – Tokyo, 2003. – 13 p.
4. Zhou, D., Translation techniques in cross-language information retrieval. / D. Zhou, M. Truran, T. Brailsford - ACM Comput. Surv. 2012. - 44 p.
5. Лукашевич, Н. В. Тезаурусы в задачах информационного поиска. / Н. В. Лукашевич - МГУ, 2011. - 495 с.
6. Voorhees, E. Using WordNet for Text Retrieval. / E. Voorhees – MIT Press, 1998. – 185-304 p.
7. Leacock, C. Combining local context and WordNet similarity for word sense identification / C. Leacock, M. Chodorow – MIT Press., 1998. – 265-283 p.
8. Azzini, A. Evolving Neural Networks for Word Sense Disambiguation: 8th International Conference on hybrid intelligent systems. Spain. Barcelona, 2008 / A. Azzini, M. Dragoni - Barcelona, 2008. - 332-337 p.
9. Ciaramita, M. Explaining away ambiguity: Learning verb selectional preference with Bayesian networks: In Proceedings of the 18th Conference on Computational linguistics, Saarbrücken, Germany 2000 / M. Ciaramita, M. Johnson - Saarbrücken, 2000. - 187–193 p.
10. Pedersen, T. A Simple Approach to Building Ensembles of Naive Bayesian Classifers for Word Sense Disambiguation / T. Pedersen - Department of Computer Science, University of Minnesota Duluth, 2000. - 7 p.
11. Banerjee, S. An adapted Lesk algorithm for word sense disambiguation using WordNet: In Proceedings of the Third International Conference on Intelligent Text Processing and Computational Linguistics, Mexico City, February 2002 / S. Banerjee, T. Pedersen - Mexico, 2002. - 9 p.