Министерство образования республики беларусь

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

**Факультет прикладной математики и информатики**

**Кафедра информационных систем управления**

**CROSS-LANGUAGE ФУНКЦИОНАЛЬНОСТЬ АВТОМАТИЧЕСКОГО ПОИСКА В СЕТИ INTERNET РЕЛЕВАНТНЫХ ДОКУМЕНТОВ**

Отчёт по преддипломной практике

|  |  |
| --- | --- |
|  | Исаченко Дмитрия Александровича  студента 5 курса,  специальность «информатика» |
|  | Научный руководитель:  доктор технических наук,  профессор И.В. Совпель |

Минск 2017

**РЕФЕРАТ**

Отчёт по преддипломной практике, 24 стр., 8 рис., 9 источников.

Объектом исследования являются системы, позволяющие для произвольного документа обнаружить релевантные ему документы в сети интернет, в том числе на отличном от исходного языка.

Цель работы: исследовать решения поиска текстовых документов релевантных данному, разработать соответствующий алгоритм и, в соответствии с ним, реализовать приложение.

Результатом работы является приложение под платформу Android, взаимодействующее с облачным хранилищем и позволяющие для произвольного документа обнаружить на заданных пользователем языках релевантные ему документы в сети интернет.

Область применения результатов: классификация и анализ текстов, информационный поиск.

ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 6](#_Toc481432228)

[ГЛАВА 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ И ПОСТАНОВКА ЗАДАЧИ 7](#_Toc481432229)

[1.1 Исследование подходов поиска релевантных документов в одноязычной информационной среде 7](#_Toc481432230)

[1.1.1 Реализация собственной поисковой системы 7](#_Toc481432231)

[1.1.2 Составление поискового образа документа 10](#_Toc481432232)

[1.1.3 Методы извлечения ключевых слов 11](#_Toc481432233)

[1.2 Решение задачи в многоязычной информационной среде 15](#_Toc481432234)

[1.2.1 Электронный тезаурус Wordnet 17](#_Toc481432235)

[1.3 Разрешение лексической многозначности слов. 20](#_Toc481432236)

[1.3.1 Методы, основанные на использовании тезаурусных знаний. 21](#_Toc481432237)

[1.3.2 Методы на основе нейронных сетей, построенных по данным машиночитаемых словарей 22](#_Toc481432238)

[1.3.3 Бустинг 23](#_Toc481432239)

[1.3.4 Использование лексических цепочек для разрешения многозначности. 24](#_Toc481432240)

[1.3.5 Разрешение лексической многозначности методом ансамбля байесовских классификаторов. 25](#_Toc481432241)

[1.3.6 Контекстная кластеризация 26](#_Toc481432242)

[1.3 Постановка задачи 27](#_Toc481432243)

[1.4 Выводы 27](#_Toc481432244)

[ГЛАВА 2. АЛГОРИТМИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ЗАДАЧИ 28](#_Toc481432245)

[2.1 Описание алгоритма 28](#_Toc481432246)

[2.2 Стемминг 28](#_Toc481432247)

[2.2.1 Алгоритмы поиска 29](#_Toc481432248)

[2.2.2 Алгоритмы усечения окончаний 29](#_Toc481432249)

[2.2.3 Алгоритмы лемматизации 30](#_Toc481432250)

[2.2.4 Алгоритмы стемминга на основе корпуса текстов 30](#_Toc481432251)

[2.2.5 Алгоритмы сопоставления 30](#_Toc481432252)

[2.3 Определение важности слова в контексте документа 31](#_Toc481432253)

[2.4 Выводы 32](#_Toc481432254)

[ГЛАВА 3. РЕАЛИЗАЦИЯ СИСТЕМЫ 33](#_Toc481432255)

[3.1 Разработка архитектуры системы 33](#_Toc481432256)

[3.2 Особенности реализации мобильного приложения 33](#_Toc481432257)

[3.3 Методика применения разработанного приложения 34](#_Toc481432258)

[3.4 Пример использования разработанной системы 34](#_Toc481432259)

[3.5 Выводы 37](#_Toc481432260)

[ЗАКЛЮЧЕНИЕ 38](#_Toc481432261)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 39](#_Toc481432262)

**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** - это процесс нахождения основы слова для заданного исходного слова. Основа слова необязательно совпадает с морфологическим корнем слова;

**TF (term frequency)** – частота термина в контексте определённого документа;

**IDF** (**inverse document frequency**) - обратная частота термина в корпусе документов;

**TF-IDF** - статистическая мера, используемая для оценки важности слова в контексте документа, являющегося частью коллекции документов или корпуса;

**CLIR** - разновидность информационного поиска, при которой язык извлечённой информации может отличаться от языка запроса.

# ВВЕДЕНИЕ

В наше время огромное количество информации, в том числе текстовые документы, доступны в электронном виде. Информационные системы, оперирующие большими объемами текстовых документов произвольной предметной области и успешно решающие различные прикладные задачи, становятся все более востребованными как предприятиями и организациями, так и отдельными пользователями. При этом обработка информации, представленной в документах на различных языках, не является тривиальной. В связи с этим актуальна задача автоматизации поиска в сети интернет документов релевантных данному, в том числе на отлихных от иходного языках, что позволяет получить максимальное количество различной информации по теме исходного документа. Эту задачу можно свести к формированию поискового запроса, максимально описывающего тему данного документа, что в свою очередь сводится к задаче определения ключевых слов в тексте. Существуют следующие категории методов выделения ключевых слов: статистические, лингвистические и гибридные, которые являются их комбинацией. Популярные статистические методы для измерения важности слов используют статистическую меры TF-IDF, которая учитывает, как часто данное слово встречается в документе и в то же время как редко – в корпусе документов. Такой способ нахождения ключевых слов обладает более высокой точностью по сравнению с другими, в которых не задействуется предварительное обучение системы на корпусе документов.

В данной работе описывается подход по формированию поискового запроса для автоматизации поиска релевантных данному документов, в том числе на отлихных от иходного языках. В связи с тем, что большинство людей сейчас проводят больше времени в своих мобильных телефонах/планшетах, чем в десктопах, приложение будет разрабатываться под платформу Android на языке программирования Java.

# ГЛАВА 1. Поиск релевантных документов в одноязычной информационной среде

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

* реализовать собственную поисковую систему при разработке приложения;
* воспользоваться уже существующей поисковой системой при разработке приложения.

## 1.1 Реализация собственной 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, которая предназначена для перебора страниц Интернета с целью занесения информации о них в базу 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. Владельцы поисковых машин нередко ограничивают глубину проникновения внутрь сайта и максимальный размер сканируемого текста, поэтому чересчур большие сайты могут оказаться не полностью проиндексированными поисковой машиной. Порядок обхода страниц, частота визитов, защита от зацикливания, а также критерии выделения значимой информации определяются алгоритмами информационного поиска. В большинстве случаев переход от одной страницы к другой осуществляется по ссылкам, содержащимся на первой и последующих страницах. Также многие поисковые системы предоставляют пользователю возможность самостоятельно добавить сайт в очередь для индексирования. Обычно это существенно ускоряет индексирование сайта, а в случаях, когда никакие внешние ссылки не ведут на сайт, вообще оказывается практически единственной возможностью указать на его существование.

Индексатор — это модуль, который анализирует страницу, предварительно разбив её на части, применяя собственные лексические и морфологические алгоритмы. Все элементы веб-страницы вычленяются и анализируются отдельно. Данные о веб-страницах хранятся в индексной базе данных для использования в последующих запросах. Индекс позволяет быстро находить информацию по запросу пользователя.

Индексирование в поисковых системах — процесс добавления сведений роботом поисковой машины в базу данных, впоследствии использующуюся для поиска информации на проиндексированных сайтах. В сведения о сайте чаще всего входят ключевые слова (алгоритм определения ключевых слов зависит от поисковой системы), статьи, ссылки, документы, также могут индексироваться изображения, аудио и т. д.

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 поисковый запрос представляет собой набор слов или фразы.

Поисковые системы работают поэтапно. Сначала поисковый робот получает контент, затем индексатор генерирует доступный для поиска индекс, и наконец, поисковик обеспечивает функциональность для поиска индексируемых данных. Чтобы обновить поисковую систему, этот цикл индексации выполняется повторно.

Поисковые системы работают, храня информацию о многих веб-страницах, которые они получают из HTML страниц. Поисковая система анализирует содержание каждой страницы для дальнейшего индексирования. Слова могут быть извлечены из заголовков, текста страницы или специальных полей — метатегов. Ряд поисковых систем, подобных Google, хранят исходную страницу целиком или её часть, так называемый кэш, а также различную информацию о веб-странице. Другие системы, подобные системе AltaVista, хранят каждое слово каждой найденной страницы. Использование кэша помогает ускорить извлечение информации с уже посещённых страниц. Кэшированные страницы всегда содержат тот текст, который пользователь задал в поисковом запросе. Это может быть полезно в том случае, когда веб-страница обновилась, то есть уже не содержит текст запроса пользователя, а страница в кэше ещё старая. Поисковик работает с выходными файлами, полученными от индексатора. Поисковик принимает пользовательские запросы, обрабатывает их при помощи индекса и возвращает результаты поиска. Когда пользователь вводит запрос в поисковую систему (обычно при помощи ключевых слов), система проверяет свой индекс и выдаёт список наиболее подходящих веб-страниц обычно с краткой аннотацией, содержащей заголовок документа и иногда части текста. Поисковый индекс строится по специальной методике на основе информации, извлечённой из веб-страниц. Полезность поисковой системы зависит от релевантности найденных ею страниц. Хоть миллионы веб-страниц и могут включать некое слово или фразу, но одни из них могут быть более релевантны, популярны или авторитетны, чем другие. Большинство поисковых систем использует методы ранжирования, чтобы вывести в начало списка «лучшие» результаты. Поисковые системы решают, какие страницы более релевантны, и в каком порядке должны быть показаны результаты. Методы поиска, как и сам Интернет со временем меняются. Так появились два основных типа поисковых систем: системы предопределённых и иерархически упорядоченных ключевых слов, и системы, в которых генерируется инвертированный индекс на основе анализа текста.

В связи с огромной трудоёмкостью реализации собственной поисковой системы при разработке приложения 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 документа, который формируется из ключевых для исходного текста слов. Количество слов в запросе можем варьироваться в зависимости от размера документа. Согласно рекомендациям Google, поисковой запрос должен состоять из ключевых слов, оптимальное количество ключевых слов должно находиться в диапазоне 6-8.

## 1.2 Составление поискового образа документа

Поисковый образ документа - 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 выделения ключевых словосочетаний используется анализ коллокаций, которые выявляются в ходе лексического анализа текста.

Коллокация – словосочетание, состоящее из двух или более слов, имеющее признаки синтаксически и семантически целостной единицы, в котором выбор одного из компонентов осуществляется по смыслу, а выбор второго зависит от выбора первого. Для обнаружения коллокаций используются различные меры ассоциативной связи, которые оценивают, является ли взаимное появление лексических единиц случайным, или оно статически значимо.

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

### 1.2.1 Предварительная обработка документа перед составлением ПОД

Сначала для исходного текста выделяются границы слов в ходе процесса токенизации. Затем для обнаружения различных форм одного и того же слова необходимо выполнить процесс нахождения основы слова для всех исходных слов текста – стемминг. Стемминг выполняет морфологический разбор слова, находит общую для всех его грамматических форм основу, отсекая суффиксы и окончания. Одним из самых распространённых стеммеров для английского языка является стемминг Портера в силу его быстродействия, отсутствия необходимости в предварительной обработке корпуса документов и использования каких-либо баз основ.

Основная идея стеммера Портера заключается в том, что существует ограниченное количество словообразующих суффиксов и вручную заданы некоторые правила. Алгоритм состоит из пяти шагов. На каждом шаге отсекается словообразующий суффикс, и оставшаяся часть проверяется на соответствие правилам. Если полученное слово удовлетворяет правилам, происходит переход на следующий шаг. Если нет — алгоритм выбирает другой суффикс для отсечения. Минусами данного алгоритма является то, что он часто обрезает слово больше необходимого, а это затрудняет получение правильной основы слова. Также стеммер Портера не справляется со всевозможными изменениями корня слова (например, выпадающие и беглые гласные).

В зависимости от необходимой производительности и точности можно выбрать алгоритм стемминга из приведенных ниже типов.

Простой стеммер ищет флективную форму в таблице поиска. Преимущества этого подхода заключается в его простоте, скорости, а также легкости обработки исключений. К недостаткам можно отнести то, что все флективные формы должны быть явно перечислены в таблице: новые или незнакомые слова не будут обрабатываться, даже если они являются правильными (например, iPhones ~ iPhone), а также проблемой является то, что таблица поиска может быть очень большой.Для языков с простой морфологией наподобие английского размеры таблиц небольшие, но для сильно флективных языков (например, турецкий) таблица может иметь сотни возможных флективных форм для каждого корня.

Таблицы поиска, используемые в стеммерах, как правило, генерируются в полуавтоматическом режиме. Например, для английского слова «run» автоматически будут сгенерированы формы «running», «runs», «runned» и «runly». Последние две формы являются допустимыми конструкциями, но они вряд ли появятся в обычном тексте на английском языке.

Алгоритм поиска может использовать предварительную частеречную разметку, чтобы избежать такой разновидности ошибки лемматизации, когда разные слова относят к одной лемме

Алгоритмы усечения окончаний не используют справочную таблицу, которая состоит из флективных форм и отношений корня и формы. Вместо этого, как правило, хранится меньший список «правил», который используется алгоритмами, учитывая форму слова, чтобы найти его основу. Некоторые примеры правил выглядят следующим образом:

* если слово оканчивается на 'ed', удалить 'ed'
* если слово оканчивается на 'ing', удалить 'ing'
* если слово оканчивается на 'ly', удалить 'ly'

Алгоритмы усечения окончаний неэффективны для исключительных ситуаций (например, 'ran' и 'run')). Решения, полученные алгоритмами усечения окончаний, ограничиваются теми частями речи, которые имеют хорошо известные окончания и суффиксы с некоторыми исключениями. Это является серьезным ограничением, так как не все части речи имеют хорошо сформулированный набор правил. Лемматизация пытается снять это ограничение.

Более сложным подходом к решению проблемы определения основы слова является лемматизация. Чтобы понять, как работает лемматизация, нужно знать, как создаются различные формы слова. Большинство слов изменяется, когда они используются в различных грамматических формах. Конец слова заменяется на грамматическое окончание, и это приводит к новой форме исходного слова. Лемматизация выполняет обратное преобразование: заменяет грамматическое окончание суффиксом или окончанием начальной формы.

Также лемматизация включает определение части речи слова и применение различных правил нормализации для каждой части речи. Определение части речи происходит до попытки найти основу, поскольку для некоторых языков правила стемминга зависят от части речи данного слова.

Одим из главных недостатков классических стеммеров (например, стеммер Портера) является то, что они часто не различают слова со схожим синтаксисом, но совершенно с разными значениями. Например, «news» и «new» в результате стемминга сведутся к основе «new», хотя данные слова принадлежат к разным лексическим категориям. Другая проблема заключается в том, что некоторые алгоритмы стемминга могут быть пригодны для одного корпуса и вызывать слишком много ошибок в другом. Например, слова «stock», «stocks», «stocking» и т. д. будут иметь особое значение в текстах газеты The Wall Street Journal. Основная идея стемминга на основе корпуса текстов состоит в создании классов эквивалентности для слов классических стеммеров, а затем «разбить» некоторые слова, объединенные на основе их встречаемости в корпусе.

Такие алгоритмы используют базу данных основ (например, набор документов, содержащие основы слов). Данные основы не обязательно соответствуют обычным словам, в большинстве случаев основа представляет собой подстроку (например, для английского языка «brows» является подстрокой в словах «browse» и «browsing»). Для того, что определить основу слова алгоритм пытается сопоставить его с основами из базы данных, применяя различные ограничения, например, на длину искомой основы в слове относительно длины самого слова (так, например, короткий префикс «be», который является основой таких слов, как «be», «been» и «being», не будет являться основой слова «beside»).

## 1.3 Методы извлечения ключевых слов

Существуют следующие категории методов выделения ключевых слов: статистические, лингвистические, и гибридные, которые являются их комбинацией.

### 1.3.1 Лингвистические методы извлечения ключевых слов.

В основе лингвистических методов лежат значения слов, также используются онтологии (попытки формализации некоторой области знаний с помощью концептуальной схемы) и семантические данные о слове. Эти методы слишком трудоемки на ранних этапах: разработка онтологий является очень затратным процессом. К тому же, операции лингвистического анализа текстов, выполняемые вручную, являются источником значительного количества ошибок и неточностей и делают сам процесс анализа документов сложным и длительным. Поэтому необходимым условием является наличие доступных программных средств, позволяющих автоматизировать процесс анализа текстов документов.

Лингвистические методы, основанные на графах, представляют большой интерес в области обработки естественного языка. В этих методах основной процедурой является построение семантического графа. Семантический граф - взвешенный граф, вершинами которого являются термины документа, две вершины связанны ребром тогда и только тогда, когда термины семантически связаны между собой, численное значение семантической близости двух терминов - вес ребра. Ключевые слова отбираются алгоритмами обработки графа. Графовые методы различаются между собой способами отбора множества терминов и определения близости отдельных терминов, которые основаны на статистических параметрах, а также на морфологическом, синтаксическом или семантическом анализе.

### 1.3.2 Статистические методы извлечения ключевых слов.

Статистические методы базируются на численных данных о встречаемости слова в тексте. Их преимуществами являются универсальность алгоритмов извлечения ключевых слов, отсутствие необходимости в трудоемких процедурах построения лингвистических баз знаний, а также относительная простота реализации. 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 (для выделения ключевых слов), а также анализ коллокаций (для выделения словосочетаний). Одним из элементарных статистических методов извлечения ключевых слов является построение множества кандидатов путем ранжирования всех словоформ или лексем документа по частоте. Фильтрация в данном случае осуществляется через отбор в качестве ключевых наиболее частотных словоформ/лексем.

При использовании частоты слова в документе в качестве единственного параметра для автоматического извлечения ключевых слов подсчет общей частоты словоформ из парадигмы одной лексемы чаще всего осуществляется следующим образом: общая частота ключевых слов подсчитывается путем сравнения словоформ, нормализованных к одной форме, как правило, к основе или лемме. Автоматическая нормализация словоформы является задачей морфологического анализа, которая достаточно проблематична сама по себе. При статистических подходах к извлечению ключевых слов используются простые эвристические алгоритмы, чаще всего нормализующие словоформу к ее квази-основе, отсекая от словоформы определенное количество букв. Такие алгоритмы носят название стемминг-алгоритмы, наиболее известным из них является стемминг-алгоритм Портера. Нормализованные словоформы ранжируются по частоте и те из них, чья частота выше заданного порога, считаются ключевыми. Использование корпуса текстов для повышения корректности извлечения ключевых слов получило достаточно широкое распространение, однако отсутствие таких корпусов для каждой конкретной предметной области в реальной жизни делает применение таких корпусных моделей и методов весьма проблематичным. Для более точного представления содержания документа разрабатываются методики, использующие в качестве меры релевантности вес лексемы, складывающийся из некоторой комбинации значений различных параметров лексем, например, частоты лексемы в документе, расположения в определенной части текста (например, в заголовке или начале параграфа), статистики совместной встречаемости слов в документе/корпусе и их дисперсии или отношения логарифмического правдоподобия.

Преимуществами чисто статистического подхода являются универсальность алгоритмов извлечения ключевых слов и отсутствие необходимости в трудоемких и время затратных процедурах построения лингвистических баз знаний. Несмотря на указанные преимущества статистических методов извлечения ключевых слов, чисто статистические методы часто не обеспечивают удовлетворительного качества результатов. При этом область их применения ограничена языками с бедной морфологией, такими как английский, где частотность словоформ одной лексемы велика. Чисто статистические модели извлечения ключевых слов, удовлетворительно работающие, например, на материале английского языка, не пригодны для естественных языков с богатой морфологией, в частности, для русского языка, где каждая лексема характеризуется большим количеством словоформ с низкой частотностью в каждом конкретном тексте.

Для оценки важности слова в контексте документа будем использовать статистическую меру 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.

Введём следующие обозначения:

* D - корпус документов;
* N - размер корпуса документов;
* t - термин, важность которого хотим определить в документе d.

Тогда:

n(t) = 1 + количество документов, в которых встречается термин t;

f(t,d) = количество раз, которое термин t встречается в документе d.

Способы определения статистики TF:

* по частоте встречаемости (raw frequency): ;
* логический (boolean frequency): ;
* логарифмически нормализованный (logarithmically scaled frequency): ;
* нормализованный по максимальной частоте слова (augmented frequency): .

Cпособы определения статистики IDF:

* ;
* ;
* ;
* .

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.

### 1.3.3 Гибридные методы извлечения ключевых слов.

Для повышения корректности автоматического извлечения ключевых слов используются гибридные методики, в которых статистические методы обработки документов дополняются одной или несколькими лингвистическими процедурами (морфологическим, синтаксическим, и семантическим анализами) и лингвистическими базами знаний различной глубины (словарями, онтологиями, грамматиками, лингвистическими правилами и т. д.). Гибридные методы извлечения ключевых слов из документа, также как и статистические, могут требовать или не требовать корпуса текстов. Например, метод Кена Баркера, включает поиск в тексте документа базовых именных групп с использованием морфосинтаксического анализа на основе словарей и вычисление релевантности БИГ. Ключевыми считаются именные группы с показателем релевантности выше заданного порога. К гибридным методам извлечения ключевых слов можно отнести методы на основе машинного обучения, где задача извлечения ключевых слов рассматривается как задача классификации. Методы на основе машинного обучения для создания обучающей выборки и построения модели-классификатора, как правило, требуют корпуса документов с размеченными ключевыми словами. Помеченные ключевые слова считаются положительным примером, остальные слова – отрицательным примером. Далее высчитывается релевантность каждого слова тренировочного текста путем сопоставления ему вектора значений различных параметров, например, меры TF-IDF, длины слова, части речи, положения слова в заголовке, положения слова в первом абзаце, последнем абзаце, в списках литературы и т. д. Фиксируются отличие значений векторов этих параметров для ключевых слов и не ключевых. Далее вычисляется вероятность отнесения каждого слова к группе ключевых и задается ее порог, т. е. модель обучается. Извлечение ключевых слов из нового документа происходит путем вычисления релевантности слов и их вероятности отнесения к ключевым в соответствии с построенной моделью.

Проанализировав вышеописанные методы, было замечено, что общая схема извлечения ключевых слов из текста практически одинакова для всех используемых методов и состоит из следующих этапов:

1. Предварительная обработка текста, призванная представить текст в формате, удобном для последующего распознавания. Она включает в себя: удаление 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 – Типовая последовательность этапов извлечения ключевых слов**

В связи с трудоёмкостью реализации собственного лингвистического процессора в данной работе для выделения ключевых слов при формировании поискового запроса будет рассмотрен статистический метод, использующий статистическую меру TF-IDF.

# Глава 2. Поиск релевантных документов в многоязычной информационной среде

При разработке собственной поисковой системы, поддерживающей обнаружение информации на языке отличной от языка запроса, можно было бы перевести все имеющиеся документы на все возможные языки запросов.



**рисунок 1.2 – CLIR с переводом документов**

Так же можно было бы ввести промежуточный язык, на который бы переводились все документы и поисковой запрос.

**рисунок 1.3 – CLIR с переводом документов и запроса на промежуточный язык**

У каждого из данных подходов имеются свои плюсы и минусы, но так как мы решили не разрабатывать собственную поисковую систмему, а воспользоваться уже существующей, то рассмотрим третий подход, основывающийся на переводе запроса.

**рисунок 1.4 – CLIR с переводом запроса**

В данном случае запросом является исходный текст, процесс индексации – процесс построения поискового образа документа, а затем с учётом того, что в поисковой системе Google язык результатов запроса тот же, что и язык исходного запроса, необходимо выполнить машинный перевод на целевой для результатов язык.

Алгоритм составления ПОД для документов на различных языках может отличаться, в силу особенностей языков. Например, в Китайском языке нету пробельных разделителей и в связи с этим для использования статистических методов необходимо предварительно выделить границы слов в исходном тексте.

**//TODO добавь что-нибудь про выделение границ слов**

Таким образом для исходного текста сначала будем составлять ПОД, а затем выполнять его перевод. Для повышения точности перевода будем использовать тезаурус синсетов.

## 2.1 Электронные тезаурусы. Лексическая база данных Wordnet.

Тезаурус – словарь, охватывающие понятия, определения и термины специальной области знаний. Слова в тезаурусах упорядочены по смысловой близости, не по алфавиту.

Наиболее распространёнными типами смысловых отношений между словами в тезаурусах являются:

* синонимия, базирующаяся на критерии, что два выражения являются синонимичными, если замена одного из них на другое в предложении не меняет значения истинности этого высказывания, 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. При работе со словом учитываются все его значения, особенно те, в которых это слово является синонимом к другим словам. Многозначные слова, рассматриваемые в разных значениях, входят и в разные синсеты: золотая (монета) – сделанная из золота и золотой (работник) – хороший.

WordNet - это огромная лексическая база знаний для английского языка. WordNet является семантической сетью, узлы которойпредставляют собой синсеты, связанные различными отношениями, такими как гипонимия, гиперонимия, голонимия, меронимия и т.п

WordNet приобрёл популярность благодаря его содержательными 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. В состав тезауруса входят лексемы, относящиеся к четырем частям речи: прилагательное, существительное, наречие и глагол. Лексемы различных частей речи хранятся отдельно, и описания, соответствующие каждой части речи, имеют различную структуру.

Существительные, прилагательные, глаголы, наречия сгруппированы в наборы когнитивных синонимов (синсеты), каждый из которых выражает отдельное значение. Синсеты взаимосвязаны между собой посредством концептуально-семантических и лексических отношений. WordNet является свободно распространяющейся и соответственно общедоступной для загрузки базой знаний. Тем самым структура WordNet делает его полезным инструментом для вычислительной лингвистики и обработки естественного языка.

Существительные, прилагательные, глаголы, наречия сгруппированы в наборы когнитивных синонимов (синсеты), каждый из которых выражает отдельное значение. Синсеты взаимосвязаны между собой посредством концептуально-семантических и лексических отношений. WordNet является свободно распространяющейся и соответственно общедоступной для загрузки базой знаний. Тем самым структура WordNet делает его полезным инструментом для вычислительной лингвистики и обработки естественного языка.

Слова в WordNet группируются вместе на основе их значений. WordNet внешне напоминает тезаурус, однако есть некоторые важные различия. Во-первых, WordNet связывает не только словоформы, но и слова со схожим смыслом. Во-вторых, WordNet отмечает семантические отношения между словами, тогда как группировки слов в тезаурусе не следуют какой-либо явной схеме, кроме сходства.

Основным отношением между словами в WordNet является синонимия. Синонимы - слова, которые обозначают одну и ту же концепцию и являются взаимозаменяемыми во многих контекстах. Они группируются в неупорядоченные наборы (синсеты). Каждый из 117 000 синсетатов 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, иначе в естественном языке было бы слишком мало синонимов. Для существительных в WordNet установлены 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. Все иерархии существительных в конечном счете поднимаются на корневой узел. 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 гаоборот Синсет А – гипероним синсета В, в том случае, когда существуют предложения типа А имеет разновитность В.

Меронимия или другими словами отношение «часть-целое» имеет место между синсетами, такими как, например, стул и спинка, стул и ножки. В WordNet выделяются 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 являются естественными для А и В, интерпретируемых как родовые понятия.

Прилагательные подразделяются на относительные и качественные. В связи с тем, что человек при описании качественного прилагательного зачастую называет антонимы для него, семантическое описание качественных прилагательных основано на отношении антонимии. Пары «прямых» антонимов, таких как, например, сухой и мокрый, пожилой и молодой отражают смысловое значение друг друга. Таким образом семантически связанные прилагательные являются «косвенными антонимами». В отдельную группу прилагательных выносятся прилагательные оттенков цветов. Значение относительных прилагательных является отсылкой к соответствующему синсету существительных, таким образом относительные прилагательные указывают на существительные от которых они образуются.

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, ухода за телом и глаголы, относящиеся к социальному поведению. 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 А и В, если из предложения “…выполняется А" следует, что “… выполняется В”. Например, из того, что "Девушка говорит", следует, что "Девушка издаёт звуки".

Для установления иерархических отношений между глаголами было введено отношение тропонимии. То есть "… делать А есть делать В в определённой форме". Например, “Шептать – это тихо разговаривать". Отношение тропонимии – особый вид отношения следования. Отношение причины связывает два глагольных синсета, один из которых называется результатив, а второй каузатив. Отношение причины также может быть рассмотрено как специальный случай следования. Если А влечёт за собой В, то из В также логически следует А.

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

## 2.2 Разрешение лексической многозначности слов.

Под неоднозначностью/многозначностью языкового выражения понимают наличие у него одновременно нескольких различных смыслов. Многозначность подразделяется на следующие типы: лексическую, синтаксическую и речевую, однако термин «WSD» включает в себя разрешение именно лексической (смысловой). Например, слово “ключ” может употребляться в одном из следующих значений: ключ как инструмент для открывания и ключ как источник воды.

Процесс разрешения требует нескольких вещей: системы словарных знаний для определения множества значений слов и корпус текстов для разрешения. Знания являются одними из ключевых моментов разрешения многозначности: они предоставляют данные, на которые опирается сам процесс разрешения. Эти данные могут быть как корпусы текстов, так и словари, тезаурусы, глоссарии, онтологии и т. д.

Среди основных методов разрешения лексической многозначности выделяют: методы, использующие внешние источники информации, и методы, базирующиеся на машинном обучении, работающие на размеченных корпусах текстов. Также применяются комбинации этих методов По другой классификации, методы разрешения лексической многозначности различают по типу используемых внешних источников информации: структурированные источники данных (машиночитаемые словари, тезаурусы, онтологии), неструктурированные источники данных в виде корпусов.

Далее будут представлены примеры методов и алгоритмов разрешения лексической многозначности, разбитые на группы:

* методы, основанные на использовании тезаурусов, словарей;
* методы, использующие нейронные сети;
* бустинг;
* лексические цепочки – построение последовательности семантически связанных слов;
* метод ансамбля байесовских классификаторов и сочетаемостные ограничения на основе байесовских сетей;
* контекстная кластеризация – кластеризация контекстных векторов, где разные кластеры соответствуют разным значениям слова.

### 2.2.1 Методы, основанные на использовании тезаурусных знаний.

В качестве примера одного из данных методов рассмотрим метод Леска, 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 этом определении смысл должен был частично совпадать со смыслом значений соседних слов в контексте.

Метод леска можно разбить на следующие шаги:

* Для исходного слова выделяется контекста, размер которого не более 10 ближайших по расположению слов;
* Для исходного слова осуществляется поиск всех определений в словаре;
* Сопоставление слов из контекста с каждым найденным определением. В случае если какое-либо из контекста слово присутствует в определении, то этому определению дается балл;
* Наиболее вероятным значением является то, определение которого набрало наибольшее количество баллов.

### 2.2.2 Методы на основе нейронных сетей, построенных по данным машиночитаемых словарей

В типичной нейронной сети на вход подается слово, значение которого требуется установить, т. е. целевое слово, а также контекст, его содержащий. Узлы выхода соответствуют различным значениям слова. В процессе обучения, когда значение тренировочного целевого слова известно, веса связующих узлы соединений настраиваются таким образом, чтобы по окончании обучения выходной узел, соответствующий истинному значению целевого слова, имел наибольшую активность. Веса соединений могут быть положительными или отрицательными и настраиваются посредством рекуррентных алгоритмов (алгоритм обратного распространения ошибки, рекуррентный метод наименьших квадратов и т. д.). Сеть может содержать скрытые слои, состоящие из узлов, соединенных как прямыми, так и обратными связями.

Целевое слово представлено узлом, соединенным активирующими связями со смысловыми узлами, представляющими все возможные значения слова, имеющиеся в словарных статьях. Каждый смысловой узел, в свою очередь, соединен активирующими связями с узлами, представляющими слова в словарной статье, соответствующей толкованию данного значения. Процесс соединения повторяется многократно, создавая сверхбольшую сеть взаимосвязанных узлов. В идеале сеть может содержать весь словарь.

При запуске сети первыми активируются узлы входного слова (согласно принятой кодировке). Затем каждый входной узел посылает активирующий сигнал своим смысловым узлам, с которыми он соединен. В результате сигналы распространяются по всей сети в течение определенного числа циклов. В каждом цикле узлы слова и его значений получают обратные сигналы от узлов, соединённых с ними. Узлы конкурирующих значений посылают взаимно подавляющие сигналы. Взаимодействие сигналов обратной связи и подавления, в соответствии со стратегией “победитель получает все”, позволяет увеличить активацию узлов-слов и соответствующих им правильных узлов-значений, одновременно уменьшая активацию узлов, соответствующих неправильным значениям. После нескольких десятков циклов сеть стабилизируется в состоянии, в котором активированы только узлы-значения с наиболее активированными связями с узлами-словами. При обучении сети используется метод обратного распространения (back propagation).

### 2.2.3 Бустинг

Бустинг – это общий и доказуемо эффективный метод получения очень точного правила предсказания путем комбинирования грубых и умеренно неточных эмпирических правил.

Рассмотрим бустинг на примере алгоритма AdaBoost. AdaBoost является адаптивным алгоритмом, поскольку он может адаптироваться к уровням ошибок отдельных слабых гипотез. На вход алгоритма поступает обучающая выборка, где каждый элементпринадлежит некоторому домену или признаковому пространству 𝑋 и каждая метка принадлежит некоторому набору меток 𝑌. Для каждого обучающего примера 𝑖 вес распределения для целых 𝑡 обозначается, где 𝑡 – это шаг алгоритма. За начальное распределение весов принимается. Пусть метки принимают значения из множества 𝑌 = {−1, 1}. Далее на каждом шаге 𝑡, где 𝑡 = 1…𝑇, выполняется обучение с использованием текущего распределения, после чего строится слабая гипотеза  :𝑋 → {−1; 1} с ошибкой первого рода , по которой выбирается уровень значимости  и строится новое распределение для следующего шага: .

Конечная гипотеза 𝐻(𝑥) – это среднее из большинства решений 𝑇 слабых гипотез, где  – вес, присвоенный гипотезе : 

Идея алгоритма заключается в определении набора весов для обучающей выборки. Первоначально все веса примеров устанавливаются равными, но в каждом цикле веса неправильно классифицированных по гипотезе  примеров увеличиваются. Таким образом получаются веса, которые относятся к сложным примерам. Основное теоретическое свойство

AdaBoost – это способность алгоритма уменьшать ошибку обучения. AdaBoost обладает определенными преимуществами. Его быстро и просто запрограммировать. Он не имеет никаких параметров для настройки, за исключением количества циклов. Он не требует никаких предварительных знаний о слабом обучаемом и поэтому может быть скомбинирован с любым методом для нахождения слабых гипотез. Недостатки метода заключаются в следующем: фактическая производительность бустинга на конкретной задаче явно зависит от данных и слабо обучаемого алгоритма. Теоретически бустинг может выполниться плохо, если данных недостаточно, слабые гипотезы слишком сложные или, наоборот, слишком слабые. Также бустинг особенно восприимчив к шуму.

### 2.2.4 Использование лексических цепочек для разрешения многозначности.

Метод построения лексических цепочек включает шаги:

1. Выбирается набор слов-кандидатов на включение в цепочки (существительные и составные существительные).
2. По словарю строится список всех значений для каждого слова-кандидата.
3. Для каждого значения каждого слова-кандидата находится расстояние до каждого слова во всех уже построенных цепочках (слово в цепочке имеет строго определенное значение, задаваемое другими словами в той же цепочке). Между двумя словами есть отношение, если мало расстояние между этими словами в тексте или между значениями этих слов существует путь в тезаурусе WordNet. Выделяют три вида отношений:
   * + Extra-strong отношение существует для слов, повторяющихся в тексте. Повтор может быть на любом расстоянии от первого употребления слова.
     + Strong отношение определено между словами, связанными отношением в WordNet. Два таких слова должны находиться в окне не более семи предложений.
     + Medium-strong отношение указывается для слов, синсеты которых находятся на расстоянии больше одного вWordNet (но есть еще и дополнительные ограничения на путь между синсетами). Слова в тексте должны находиться в пределах трех предложений.
4. Слово-кандидат добавляется в цепочки, со словами которых найдена связь. Смысловая неоднозначность устраняется, в цепочку добавляется не просто слово, а его конкретное значение (благодаря выбору значения в словаре на шаге 2).

Для выбора приоритетной цепочки (для вставки слова-кандидата) отношения упорядочены так: extra-strong, strong, medium-strong. Цепочки можно выбирать жадным алгоритмом, при этом слово-кандидат попадает ровно в одну цепочку и после этого выбор уже не может быть изменен, даже если последующий текст покажет ошибочность первоначального решения. Так же приоритетную цепочку можно выбирать по следующей схеме, требующей рассмотрения всех возможных цепочек. Таким образом, будут сформированы цепочки с учетом всех возможных значений слов с последующим выбором наилучшей цепочки.

### 2.2.5 Разрешение лексической многозначности методом ансамбля байесовских классификаторов.

Наивный байесовский классификатор – это простой вероятностный классификатор на основе применения теоремы Байеса. Для различения значений учитывается совместная встречаемость слов в окне заданного размера в текстах корпуса. При разрешении лексической многозначности, представленном в виде задачи обучения с учителем, применяют статистические методы и методы машинного обучения к размеченному корпусу. В таких методах словам корпуса, для которых указано значение, соответствует набор языковых свойств.

Подход основан на объединении ряда простых классификаторов в ансамбль, который разрешает многозначность с помощью голосования простым большинством голосов. В проблеме разрешения лексической многозначности существует понятие контекста, в котором встречается многозначное слово. Этот контекст представляется в виде функции переменных, а значение многозначного слова представлено в виде классификационной переменной S. Все переменные бинарные. Переменная, соответствующая слову из контекста, принимает значение “ИСТИНА”, если это слово находится на расстоянии определенного количества слов слева или справа от целевого слова. Совместная вероятность наблюдения определенной комбинации переменных контекста с конкретным значением слова выражается следующим образом:, где и – параметры данной модели. Для оценки параметров достаточно знать частоты событий, описываемых взаимозависимыми переменными. Эти значения соответствуют числу предложений, где слово, представляемое, встречается в некотором контексте многозначного слова, упомянутого в значении 𝑆. Если возникают нулевые значения параметров, то они сглаживаются путем присвоения им по умолчанию очень маленького значения. После оценки всех параметров модель считается обученной и может быть использована в качестве классификатора.

Контекст представлен в виде bag-of-words (модель “мешка слов”). В этой модели выполняется следующая предобработка текста: удаляются знаки препинания, все слова переводятся в нижний регистр, все слова приводятся к их начальной форме (лемматизация). Контексты делятся на два окна: левое и правое. В первое попадают слова, встречающиеся слева от неоднозначного слова, и, соответственно, во второе – встречающиеся справа. Окна контекстов могут принимать 9 различных размеров: 0, 1, 2, 3, 4, 5, 10, 25 и 50 слов. Первым шагом в ансамблевом подходе является обучение отдельных наивных байесовских классификаторов для каждого из 81 возможных сочетаний левого и правого размеров окон. Наивный байесовский классификатор (𝑙, 𝑟) включает в себя слов слева от неоднозначного слова и 𝑟 слов справа. Исключением является классификатор (0, 0), который не включает в себя слов ни слева, ни справа. В случае нулевого контекста классификатору присваивается априорная вероятность многозначного слова (равная вероятности встретить наиболее употребляемое значение). Следующий шаг при построении ансамбля – это выбор классификаторов, которые станут членами ансамбля. 81 классификатор группируется в три общие категории, по размеру окна контекста. Используются три таких диапазона: узкий (окна шириной в 0, 1 и 2 слова), средний (3, 4, 5 слов), широкий (10, 25, 50 слов). Всего есть 9 возможных комбинаций, поскольку левое и правое окна отделены друг от друга. Например, наивный байесовский классификатор (1, 3) относится к диапазону категории (узкий, средний), поскольку он основан на окне из одного слова слева и окне из трех слов справа. Наиболее точный классификатор в каждой из 9 категорий диапазонов выбирается для включения в ансамбль. Затем каждый из 9 членов классификаторов голосует за наиболее вероятное значение слова с учетом контекста. После этого ансамбль разрешает многозначность путем присвоения целевому слову значения, получившего наибольшее число голосов.

Для разрешения многозначности можно так же воспользоваться построением сочетаемостных ограничений на основе байесовских сетей. Сочетаемостные ограничения – это закономерности использования глагола относительно семантического класса его параметров (субъект, объект (прямое дополнение) и косвенное дополнение. Модели автоматического построения сочетаемостных ограничений важны сами по себе и имеют приложения в обработке естественного языка. Сочетаемостные ограничения глагола могут применяться для получения возможных значений неизвестного параметра при известных глаголах. При построении предложения сочетаемостные ограничения позволяют отранжировать варианты и выбрать лучший среди них. Исследование сочетаемостных ограничений могло бы помочь в понимании структуры ментального лексикона. Системы обучения сочетаемостных ограничений без учителя обычно комбинируют статистические подходы и подходы, основанные на знаниях. Компонент базы знаний – это обычно база данных, в которой слова сгруппированы в классы.

Статистический компонент состоит из пар предикат-аргумент, извлеченных из неразмеченного корпуса. В тривиальном алгоритме можно было бы получить список слов (прямых дополнений глагола), и для тех слов, которые есть в WordNet, вывести их семантические классы. Семантическим классом называется синсет тезауруса WordNet, т.е. класс соответствует одному из значений слова. Таким образом, в тривиальном алгоритме на основе данных WordNet можно выбрать классы (значения слов), с которыми употребляются (встречаются в корпусе) глаголы.

Байесовские сети, или байесовские сети доверия (БСД), состоят из множества переменных (вершин) и множества ориентированных ребер, соединяющих эти переменные.

Такой сети соответствует ориентированный ациклический граф. Каждая переменная может принимать одно из конечного числа взаимоисключающих состояний. Пусть все переменные будут бинарного типа, т. е. принимают одно из двух значений: истина или ложь. Любой переменной А с родителями 𝐵1, ..., 𝐵𝑛 соответствует таблица условных вероятностей. Иерархия существительных в WordNet представлена в виде ориентированного ациклического графа. Синсет узла принимает значение “истина”, если глагол “выбирает” существительное из набора синонимов. Априорные вероятности задаются на основе двух предположений: во-первых, маловероятно, что глагол будет употребляться только со словами какого-то конкретного синсета, и вовторых, если глагол действительно употребляется только со словами из данного синсета(например, синсет ЕДА), тогда должно быть правомерным употребление этого глагола с гипонимами этого синсета (например, ФРУКТ).

Те же предположения, что для синсетов, верны и для употреблений слов с глаголами:

1. слово, вероятно, является аргументом глагола в том случае, если глагол употребляется с каким-либо из значений этого слова;
2. отсутствие связки глагол-синсет говорит о малой вероятности того, что слова этого синсета употребляются с глаголом.

Словам “вероятно” и “маловероятно” должны быть приписаны такие числа, сумма которых равна единице.

### 2.2.6 Контекстная кластеризация

Каждому вхождению анализируемого слова в корпус соответствует контекстный вектор. Выполняется кластеризация векторов, где разные кластеры соответствуют разным значениям слова. Алгоритмы кластеризации полагаются на дистрибутивную гипотезу, в соответствии с которой слова, употребляемые в схожих контекстах, считаются близкими по смыслу.

При решении задачи различения значений используются контекстные вектора: если целевое слово встречается в тестовых данных, то контекст этого слова представляется в виде вектора контекста. Вектор контекста - это средний вектор по векторам свойств каждого из слов контекста. Вектор свойств содержит информацию о совместной встречаемости данного слова с другими словами, этот вектор строится по данным корпуса текстов на этапе обучения.

Первоначально строится матрица совместной встречаемости слов по данным обучающего корпуса. Вектор свойств (строка матрицы) содержит информацию о совместной встречаемости данного слова с другими. После создания матрицы выполняется разделение тестовых данных, т. е. группировка примеров употреблений (фраз) с целевым словом. Каждому слову в примере употребления в тестовых данных соответствует вектор свойств из матрицы встречаемости. Средний вектор свойств по всем словам соответствует вектору контекста. Таким образом, набор тестовых данных, включающих употребление исследуемого слова, преобразуется в набор контекстных векторов, каждый из которых соответствует одному из употреблений целевого слова.

Различение значений происходит путем кластеризации контекстных векторов с помощью разделяющего или иерархического “сверху вниз” алгоритма кластеризации. Получающиеся кластеры составлены из употреблений близких по значению фраз, и каждый кластер соответствует отдельному значению целевого слова. Векторы свойств, полученные по небольшому корпусу текстов, имеют очень малую размерность (несколько сотен), что не позволяет полностью описать закономерности совместной встречаемости слов. Для решения этой проблемы векторы свойств слов расширяются содержательными словами, извлеченными из словарных толкований разных значений данного слова.

Данный метод может быть полезен при различении значений слов без учителя при небольшом количестве обучающих данных.

## 1.3 Постановка задачи

Требуется разработать мобильное приложение под платформу Android, взаимодействующее с облачным хранилищем и позволяющие для произвольного документа обнаружить на заданных пользователем языках релевантные ему документы в сети интернет.

## 1.4 Выводы

В первой главе получены следующие результаты:

* выполнен исследование предметной области;
* поставлена задача разработки системы.

# ГЛАВА 2. АЛГОРИТМИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ЗАДАЧИ

**//FIXME**

## 2.4 Выводы

Во второй главе получены следующие результаты:

* выполнен анализ существующих алгоритмов и их улучшений для извлечения ключевых слов;
* разработан алгоритм для автоматического поиска в сети интернет документов релевантных данному в том числе на отличных от исходных языков.

# ГЛАВА 3. РЕАЛИЗАЦИЯ СИСТЕМЫ

Разработанная система будет поддерживать следующие языки для исходного текста: English, French, German, Italian, Portuguese, Russian, Spanish, Swedish, язык же обнаруженных релевантных документов может быть один из 271 предоставленных здесь http://babelnet.org/stats#LanguagesandCoverage.

//FIXME

## 3.0 Описание алгоритма

Алгоритм:

1. Определить язык исходного текста;
2. Перевести исходный текст на заданные пользователем языки. Если система не обучалась для какого-то из заданных языков, то вместо данного языка перевести текст на английский.
3. Провести токенезацию переведённых текстов(возможно стемминг, в зависимости от языка);
4. Определить для токенов важность их в контексте соответствующего документа;
5. Исходя из полученных весов важности определить наборы ключевых слов;
6. Из составленных наборов ключевых слов сгенерировать поисковые запросы для Google;
7. Используя поисковые запросы, найти релевантные данному документы в сети интернет.

## 2.1 Описание алгоритма

Алгоритм:

1. Определить язык исходного текста;
2. Перевести исходный текст на заданные пользователем языки. Если система не обучалась для какого-то из заданных языков, то вместо данного языка перевести текст на английский.
3. Провести токенезацию переведённых текстов(возможно стемминг, в зависимости от языка);
4. Определить для токенов важность их в контексте соответствующего документа;
5. Исходя из полученных весов важности определить наборы ключевых слов;
6. Из составленных наборов ключевых слов сгенерировать поисковые запросы для Google;
7. Используя поисковые запросы, найти релевантные данному документы в сети интернет.

## 3.1 Разработка архитектуры системы

В качестве облачного хранилища была выбрана Firebase Realtime Database в связи с возможностью её бесплатного использования.

При разработке приложения был использован объектно-ориентированный подход. Приложение разбито на модули согласно функциональности. Имеются следующие агенты:

* Агент взаимодействия с облачным хранилищем;
* Агент пользовательского интерфейса, который преобразует команды пользователя и передаёт их соответствующим агентам;
* Агент, инкапсулирующий логику формирования поискового запроса для исходного документа;
* Агент, отвечающий за особенности работы с ОС Android.

## 3.2 Особенности реализации мобильного приложения

Так как реализованное мобильное приложение на текущий момент не взаимодействует с сервером и все полученные в результате вычислений данные хранятся на самом устройстве, то при обучении брался корпус, состоящий из 20 документов. Чтобы повысить точность и полноту результатов поисковых запросов перед проведением стемминга был добавлен этап фильтрации 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.

## 3.3 Методика применения разработанного приложения

Разработанная программа обладает интуитивно понятным интерфейсом.

Алгоритм использования приложения для поиска документов в сети интернет релевантных данному:

1. Запустить приложение;
2. Ввести текст документа/выбрать документ из локального хранилища и нажать “Продолжить”;
3. Выбрать актуальные для поиска информации языки и нажать “Продолжить”.

Так же можно просмотреть корпус документов, который использовался при обучении системы, и полученную в результате обучения системы статистику важности различных слов в различных языках.

## 3.4 Пример использования разработанной системы

Запуск программы и ввод текста для которого необходимо найти в сети интернет релевантные документы.



**рисунок 3.1 – Главный экран приложения**

По нажатию кнопки “Продолжить” появится экран выбора актуальных для пользователя языков. На данном экране отображены все поддерживаемые для поиска информации приложением языки.



**рисунок 3.2 – Экран выбор актуальных пользователю языков**

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

Экран результата запросов для поиска релевантных данному документов на выбранных языках. А так же экран результатов поиска для определённого запроса в поисковой системе Google приведены ниже.

 

**рисунок 3.3 – Результаты поиска релевантных документов**

Так же через навигационное меню можно просмотреть корпус документов, который использовался при обучении системы, и полученную в результате обучения системы статистику. Для этого в навигационном меню нужно выбрать пункт “Корпус” / “Словарь” соответственно.



**рисунок 3.4 – Навигационное меню приложения**

По клику на определённый документ на экране “Корпус” загрузится отдельный экран, позволяющий его прочесть и изучить детальную статистику по данному документу.

## 

## 3.5 Выводы

В третьей главе получены следующие результаты:

* разработана архитектура и описаны особенности реализации мобильного приложения;
* описана методика применения разработанного приложения.

# ЗАКЛЮЧЕНИЕ

В ходе проделанной работы:

1. Выполнено исследование предметной области;
2. Поставлена задача разработки системы;
3. Разработан алгоритм для автоматического поиска в сети интернет документов релевантных данному в том числе на отличных от исходного языках;
4. Разработана архитектура и описаны особенности реализации мобильного приложения;
5. Описана методика применения разработанного приложения.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Manning, C. D. Introduction to Information Retrieval. / C. D. Manning, P. Raghavan, H. Schütze. - Cambridge University Press, 2008. – 581 с.
2. Автоматическая обработка текстов на естественном языке и компьютерная лингвистика: учеб. пособие /Е.И. Большакова, Э.С. Клышински, Д.В. Ландэ [и др.]. - М.: МИЭМ, 2011. - 272 с.
3. Matsuo, Y. Keyword Extraction from a Single Document using Word Co-occurrence Statistical Information. / Y. Matsuo – Tokyo, 2003. – 13 с.
4. Turney, P.D. Learning algorithms for keyphrase extraction. Information Retrieval / P.D. Turney. - Ottawa, Ontario, Canada, 2000. – 477 с.
5. Porter, M.F. An algorithm for suffix stripping. / M.F. Porter – Cambridge, 1997. – 6 с.
6. ANDERKA, M., LIPKA, N., AND STEIN, B. 2009. Evaluating cross-language explicit semantic analysis and cross querying. In Proceedings of the 10th Cross-language Evaluation Forum Conference on Multilingual Information Access Evaluation: Text Retrieval Experiments (CLEF’09). Springer, 50–57 с.
7. Kraiij, W., Nie, J-Y., Simard, M.: Emebdding Web-based Statistical Translation Models in Cross-Language Information Retrieval. Computational Linguistics (2003) – 39 с.
8. The Cross-Language Evaluation Forum (CLEF). <http://clef-campaign.org>
9. Virga, P., Khudanpur, S.: Transliteration of proper names in cross-lingual information retrieval. In: ACL Workshop on Multilingual and Mixed Language Named Entity Recogni-tion (2003) – 8 c.