****МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ   
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ

(национальный исследовательский университет)»

**Факультет Информационные технологии и прикладная математика Кафедра 813**

**Направление подготовки 02.03.02 ФИИТ Группа 8-3ИНТ-4ДБ-035-15**

**Квалификация (степень) бакалавр**

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

**БАКАЛАВРА**

**На тему:** Распознавание символов в текстах при помощи тематического моделирования.

**Автор квалификационной работы**  Алексеев Александр Игоревич (\_\_\_\_\_\_\_\_\_\_\_)

(Фамилия, имя, отчество)

**Научный руководитель** Лемтюжникова Дарья Владимировна (\_\_\_\_\_\_\_\_\_\_\_)

(Фамилия, имя, отчество)

**Рецензент** (\_\_\_\_\_\_\_\_\_\_\_)

(Фамилия, имя, отчество)

**К з а щ и т е д о п у с т и т ь**

**Зав. кафедрой** Денисова Ирина Павловна (\_\_\_\_\_\_\_\_\_\_\_)

(Фамилия, имя, отчество)

“ 24 ”\_мая\_ 2019г.

Москва 2019г.

**РЕФЕРАТ**

Общий объем работы составляет 40 страниц, среди которых 7 рисунка, 1 таблица и 20 использованных источника.

ТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ, ПОИСК СИМВОЛОВ В ЛИТЕРАТУРЕ, МАШИННОЕ ОБУЧЕНИЕ, PYTHON

В данном исследовании рассматривается задача распознавания слов, имеющих неоднозначное и сильно зависимое от контекста смысловое значение, в дальнейшем называемых символами. Распознавание символов в художественной литературе обеспечивается взаимодействием двух компонент представленной далее структуры: тематического моделирования и базы знаний по символам

Основу работы составляет пополняемая база данных символов, обладающих различными трактовкой, а также отдельная система, связанная с каждым индивидуальным автором и тематикой. Не менее важной составляющей является учет эмоциональной составляющей слов, представляемая в численном виде, и учет частот слов.

Тематическое моделирование позволяет определить, к каким темам относится изучаемый текст и какие термины образуют каждую тему. Пусть имеется множество документов и множество тем, распределенных среди этих документов, а также множество слов, распределенных среди тем и документов. При этом документы представляются векторами частот тем, а не отдельных слов. Она также позволяет использовать поиск документов по схожей тематике по целому документу или фрагменту текста.

Дальнейшей перспективой данной работы является создание сервисов, помогающих экспертам и пользователям, и решать разнообразные задачи, связанные с поиском, аннотированием, тегированием, интерпретацией и оцениванием.

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 4](#_Toc10631417)

[ОСНОВНАЯ ЧАСТЬ 6](#_Toc10631418)

[ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 7](#_Toc10631419)

[1.1. История и особенности анализа текстовых данных 7](#_Toc10631420)

[1.2. Обработка естественного языка при помощи компьютерных средств и ее особенности 9](#_Toc10631421)

[1.3. Проблематика абстракций 12](#_Toc10631422)

[1.4 Задача распознавания символов 15](#_Toc10631423)

[ПРАКТИЧЕСКАЯ ЧАСТЬ 18](#_Toc10631424)

[2.1. Применяемые языки программирования, программные обеспечения, базы данных и инструментарий. 18](#_Toc10631425)

[2.2 Предобработка данных. 26](#_Toc10631426)

[2.3 Постановка задачи 29](#_Toc10631427)

[2.4 Описание экспериментов. 32](#_Toc10631428)

[ЗАКЛЮЧЕНИЕ 43](#_Toc10631429)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 44](#_Toc10631430)

# ВВЕДЕНИЕ

Обилие информации на сегодняшний день приводит к существенным затруднениям в её передаче и обработке. Современные технологии позволяют заметно облегчить и автоматизировать этот процесс. Так, например, многие организации уже перешли на электронный документооборот, позволяющий почти мгновенную передачу данных без дополнительных временных и ресурсных затрат. Кроме того, некоторые системы позволяют в автоматическом режиме обрабатывать документы, например, искать конкретные слова, заполнять шаблоны, формировать заявления на основе уже вводимой ранее информации.

При работе с текстовыми данными, не сформированными по шаблону, возникают проблемы. К таким проблемам можно отнести определение эмоционального окраски части текста, поиска сарказма, иронии, фразеологизмов, идиом и слов с неоднозначными и многозначными значениями. За последние несколько лет всё больше исследований проводится на данные темы, рассматривая их с разных субъективных и объективных сторон. Необходимо проанализировать такие научные работы и сформировать целостное представление относительно семантических основ слов, применяемых методов и используемого программного обеспечения.

**Практической значимостью** исследования заключается в разработке наиболее эффективного програмного обеспечения для анализа текстов и поиска в них многозначных слов, модификации уже существующих библиотек и интеграции.

**Объект исследования:** процесс создания программного обеспечения, обрабатывающего текстовую информацию с целью выделения многозначных или неоднозначных слов.

**Предмет исследования:** процесс разработки программного продукта, находящего символы в художественной литературе.

**Целью работы** является выявление и распознавание символов среди художественной литературы.

**Задачи исследования.**

1. Рассмотреть историю и особенности анализа текстовых данных.
2. Проанализировать возможности обработки естественного языка при помощи компьютерных средств.
3. Рассмотреть проблематику абстракций.
4. Рассмотреть проблематику распознавания символов.
5. Рассмотреть языки программирования и программные средства, определить эффективные для создания программы, выделяющей символы в художественных текстах.
6. Определение методов, необходимых для разработки программного продукта и их реализацию в программе.
7. Поиск, подготовка и пополнение баз данных, содержащей обработанные художественные тексты для работы программы.
8. Разработать программный продукт, реализующий поиск символов в художественных текстах конкретной базы данных.

**Гипотеза исследования:** использование программного продукта, реализующего поиск символов в художественных текстах совместно с распознаванием объектов на изображении, позволит извлекать более полную информацию из картин.

В ходе работы были использованы следующие **методы исследования**: теоретические - анализ, синтез, классификация и моделирование, эмпирические - эксперимент и сравнение.

Работа состоит из введения, основной части из двух глав и заключения, а также приложения на ... листах.

# ОСНОВНАЯ ЧАСТЬ

## ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

### 1.1. История и особенности анализа текстовых данных

Человечество давно использует письменность для своих нужд. При помощи письма велся учет имущества, регистрация и опись граждан, а также люди активно общались друг с другом. В Древнем мире, например, вели записи на глиняных табличках, пергаменте, бересте и т.д., фиксируя основные научные положения, исторические события, сохраняя память для будущих поколений. Позднее информацию восстанавливали и анализировали. На основе результатов анализа специалистами пишутся учебники и научные труды.

При этом анализ текстовых данных является важной и сложной задачей, которая с течением времени становится лишь более актуальной. Развитие информационных технологий, обширность средств массовой информации и доступность информации формирует проблему её перенасыщения. Предельное внимание уделяется автоматизированной обработке, поскольку вырастают объемы электронных носителей информации и скорость с которой она передается в условную единицу времени, что приводит к невозможности ее рациональной обработки при помощи специалистов.

В то же время, сами процессы регистрации, заполнения, сохранения и перемещения информации не стоят на месте. Активное использование специализированного программного обеспечения для документооборота привело к упрощению и ускорению бытовых и деловых взаимоотношений, в некоторой степени уменьшив влияние человеческого фактора на них. Так, продвижение в распознавание изображений по заранее заданным шаблонам позволило автоматизировать сканирование печатных и рукописных текстов, учет и выявление нарушений, и все это без участия человека.

Более того, благодаря активным исследованиям и разработкам в сфере машинного обучения, автоматизация процессов во многом изменилась. Это позволило решить следующие задачи: буквальный перевод отдельных слов и предложений; расчет прибыли; прогнозирование роста валюты и акций при учете текущей стоимости и истории изменения, и другие. Но вместе с получением первых результатов, была выявлена проблема, получившая название проблема абстракций.

Электронная вычислительная машина способна на базовом уровне анализировать цифры и буквы по отдельности или объединять их в единое значение. Однако существуют некоторые проблемы, свойственные естественным языкам. В качестве яркого примера такой ситуации можно привести проблемы, возникающие при машинном переводе текстов с одного иностранного языка на другой. В обоих языках могут быть фразеологизмы - устоявшие выражения, свойственные только для определенного языка, при этом слова, составляющие данный оборот речи, по отдельности могут даже и не иметь логической связи между друг другом. Степень эмоциональной окраски в переведенном отрезке может сильно измениться, поскольку даже обычное слово в буквальном переводе может оказаться оскорблением в другом языке. Существуют еще каламбуры - шутка, основанная на комическом использовании сходно звучащих, но разных по значению слов. При переводе передать смысл и сочетание звуков как в исходном тексте в большинстве своем невозможно. В данный список не включаются очевидные задачи, решаемые при составлении простых правил перевода, такие как учет конструктивных правил отдельных слов и целых предложений и создание прямых логических связей между словами в обоих языках.

Логично, что стандартизация текстов приводит к нивелированию большинства проблем. Это хорошо видно на примере зарубежного и отечественного документооборота, когда большая часть документов имеет строгую структуру и порядок. Также это касается и стиля написания данных документов, где часто требуется указать либо метку, либо число, а используемый эмотивный тон является нейтральным. Количество учитываемых внешних факторов сводится к минимуму, информация становится емкой, обработка как человеком, так и программным обеспечением быстрой и простой.

### 1.2. Обработка естественного языка при помощи компьютерных средств и ее особенности

Предобработка текста обязательна перед непосредственным анализом. Для этого рассмотрим представленную схему на рисунке 1.1.

Исходный текст делится на множество предложений. Дальше предложения делятся на отдельные слова, которые проходят морфологический анализ путем лемматизации или стемминга.

**Определение 1.1** Лемматизация приведения слова к нормальной форме.

**Определение 1.2** Стемминг – это процесс нахождения [основы слова](https://ru.wikipedia.org/wiki/%D0%9E%D1%81%D0%BD%D0%BE%D0%B2%D0%B0_%D1%81%D0%BB%D0%BE%D0%B2%D0%B0) для заданного исходного слова .

Делается это для того, чтобы электронной вычислительной машине было проще соотносить одно и то же слово в разных предложениях. Без данной операции слово Сталь и Стали будут считаться разными словами. Таким образом слова с разными окончаниями и предлогами приведутся к одной форме. После происходит избавление от слов, несущих малую часть информативности: союзы, предлоги, междометия.

Общая логика работы с текстом представлена на рисунке 1.1:

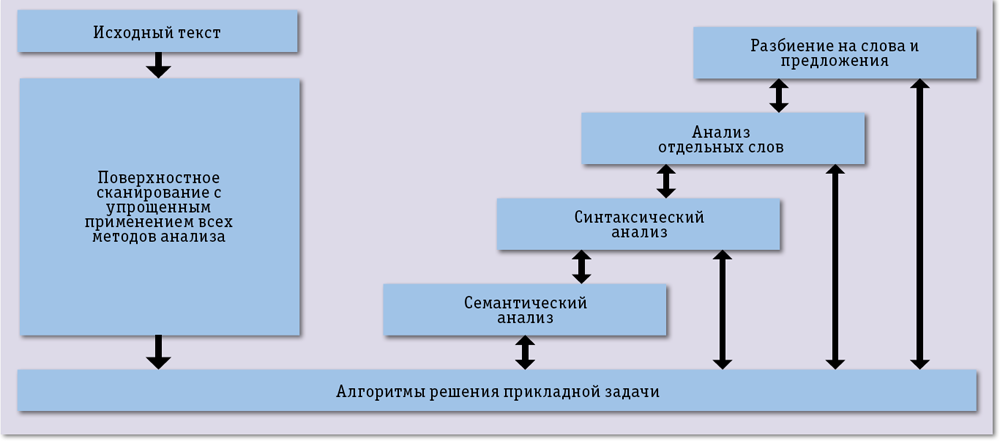


Рис. 1.1 Пример общей структуры работы с текстом

Уменьшение различий в одном слове и уменьшение количества слов является необходимым условием для следующей задачи: эмбединга слов.

**Определение 1.3.** Эмбединг слов - это приведение слов в векторное пространство путем сопоставления слова с вектором чисел, где количество чисел будет равно количеству уникальных слов в тексте, напротив ее порядкового числа стоять единица, а остальные заполняются нулями.

Так как слова в различной форме будут приведены к единой, то количество векторов будет меньшим, чем было бы без предварительной обработки.

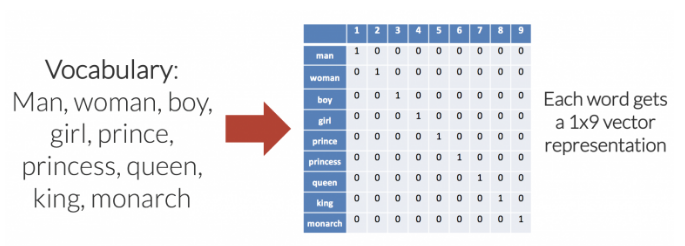


Рис. 1.2 Пример векторизации слов

Обычному обывателю будет просто сопоставить по смыслу пары слов: мужчина и женщина, мальчик и девочка, король и королева. По такой же логике работают и продвинутые модули, такие как Word2Vec, сопоставляя в матрицу векторов слов также и смысловые связи между друг другом.

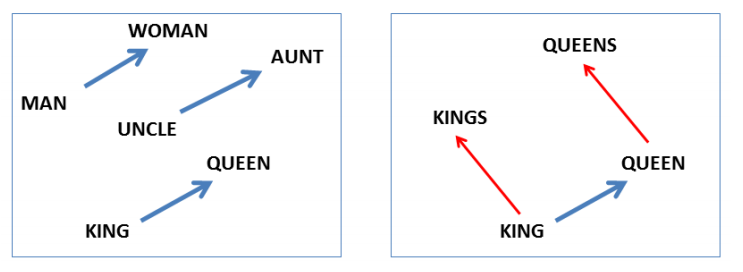


Рис. 1.3 Сопоставление семантических связей между словами

Такое взаимодействие упрощает работу со смысловой составляющей текстов, а также позволяет в некоторой степени уменьшить размер матрицы, выделенной под хранение всего словаря.

В естественном языке существуют слова, образованные от других слов и устоявшиеся в разговорной речи. Согласно [4] у таких слов смысл напрямую зависит от первоначального слова и могут быть распространены среди ограниченной аудитории. Таким образом словоформы часто наслаиваются на друг друга, вводя дополнительную многозначность. Примером является слово Трешка, значения которой могут быть как монеты номиналом в три копейки, так и квартиры с тремя комнатами. Учет таких слов приводит к расширению словаря и увеличению затрат памяти, что может быть не оправдано.

Также возникает логическая неопределенность при обработке, когда монета в три копейки была ценнее пятикопеечной монеты в силу вариативности применения. Данный аспект является крайне сложным для анализа посредством автоматизированных систем и приводит к созданию специализированных модулей. Такого же рода задачей является изучение коммуникативов, речевых штампов и репликовых частиц. Словосочетания Да ладно, Ну надо же имеют высокую частотность в разговоре, являясь каждым пятым в логической структуре предложения. Человек при их помощи выражает свое мнение: соглашается, выражает сомнение, принимает обязательства, выражает эмоциональную заинтерисованность по отношению к событиям. Чаще всего они выступают в роли ответной реакции на вопрос или происходящее. Сложность обработки возрастает, так как помимо использования единичных репликовых частиц их осложняют модальными модификаторами и грамматическими идиомами. Их наличие требует сложной предобработки текста посредством стандартизации таких словообразований до полноценных.

Производить векторизацию слов, производить разреживание массива и хранить его в памяти может быть накладно и неэффективно каждый раз. Поэтому для такой задачи привлекают лингвистов, которые составляют корпуса для языков, хранящих помимо самих слов, дополнительную информацию. Она включает пол, число и род, как дополнительные переменные, используемые в случае надобности. Туда же могут входить данные о синонимичности, гиперонимичности и гипонимичности между словами и смысловой схожести слов. Наиболее развитый корпус на текущий момент имеется в английском языке. Он включает развивающийся словарь фразеологизмов.

### 1.3. Проблематика абстракций

Основная проблематика заключается в том, что распознавание смыслов слов по их прямому значению невозможно, и, следовательно, требуется анализировать слова по некоторым иным параметрам. Проблематика распознавания смысла ставилась различными способами за последнее время. Далее взглянем на то, как ставились задачи такого типа.

В [15] и [17] рассматривается задача по выделению ключевых смысловых слов из текста. Идея состояла в выделении ключевых слов, которые часто используются по отдельности или в определённых словосочетаниях, после чего выделялись выборки ключевых слов свои ключевые слова, дополненные смыслом. В решении активно применялся наивный байесовский классификатор. Его логика заключается в применении теоремы Байеса со строгими предположениями о независимости.

В статье [14], посвященной изучению сентимент анализа, исследователи на основе изначальной разметки учили и оптимизировали алгоритм так, чтобы он мог предсказывать дальнейший эмоциональный фон в произведении на английском языке. У каждого слова имеется дополнительный признак - эмоция. Для этого используются словари тональности английского языка, в которых каждому слову ставится в соответствие величина («вес») слова относительно тональности всего текста. Далее строится функция, принимающая на вход количество вхождений в текст каждого слова, которая вычисляет агрегированную величину тональности текста целиком по следующей формуле:

где,

* – количество вхождений слова j в текст i;
* – количество слов в тональном словаре;
* – тональность текста i.

Наибольший результат был достигнут при помощи метода ближайших соседей в данном исследовании. Учет соседних слов, их семантики и эмоциональной составляющей, и смены настроения оказалось достаточно, чтобы предоставить хорошие результаты в нетривиальной задаче со стороны программирования. В дальнейшем исследователями из статьи была проделана колоссальная работа по формированию тонального словаря ассоциаций вида «слово-полярность» и «слово-эмоция» на английском языке. Рассматривались проблемы, связанные с краудсорсингом и были предложены пути их решения. В результате был получен крупный и относительно недорогой набор тональной лексики, который был переведен на ряд других языков при помощи машинного перевода.

Аналогично, авторами статьи [10] описывается методика и промежуточные результаты создания словаря тональностей русского языка для его дальнейшего использования в задачах сентимент-анализа. Сперва осуществлялась ручная разметка тональности прилагательных и наречий, основывавшаяся на гипотезе о том, что оценочные значения чаще всего выражаются в языках при помощи признаковых слов. Далее словарь был расширен и ориентирован, при помощи тематического моделирования, на тексты социальных медиа, путем вычленения характерной для данной предметной области слов. Наконец, данный словарь был вынесен на краудсорсинговую платформу, на которой в настоящее время проводится разметка эмоциональной окрашенности самих слов и отобранных текстов.

Применение машинного обучения в сфере банковского дела является уже стало его частью, так как значительно упростило, оптимизировало и автоматизировало многие процессы. Однако исследования, позволяющие улучшить существующие методы, до сих пор ведутся. Согласно [11] предобученная тематическая модель позволяет модифицировать алгоритмы для выдачи кредитов. В качестве главной темы берется кредитируемость, которая определяется на основе ранее выданных кредитов. Тематическая модель обучалась на датасетах, предоставленных компанией Yahoo. Первым датасетом выступал корпус английского языка, на котором модель определяла темы, свойственные каждой профессии. После, основная тема для кредитирования дробится в соответствии с профессией. Вторым потоком входных данных выступала история кредитования и выплат, и обезличенные данные пользователей. Качественной особенностью по сравнению с остальными алгоритмами выступал не общий показатель точности в целом, а повышение точности для узкоспециализированных профессий, так как общие модели недостаточно хорошо их классифицируют.

Популяризация социальных сетей и мессенждеров, позволившая каждому человеку свободно общаться с другим человеком или целой группой, имеющих общий интерес, обмен идеями, интересами и мыслями, стала дополнительным краеугольным камнем в передачи и накоплении данных. На конференции АИСТ были проведены исследования по тому, как влияние новостей, переговоров пользователей и работа автоматизированных ботов влияют на котировки криптовалюты. По итогу данной конференции были сделаны весьма разносторонние результаты. Например, благодаря применению метода опорных векторов совместно с нейронной сетью глубокого обучения был сделан помощник, позволяющий достаточно точно предсказывать изменения валюты на основе деловых переговоров, транзакций и графике изменения валюты. Когда как применение логистической регрессии позволило выделить пики активности ботов и коррелируемые с этим событием неожиданные скачки валюты, характерные лишь одной группе пользователей.

Платформа Kaggle [6] в январе 2019 года поставила задачу верно оценить степень сарказма в одном маленьком высказывании человека, состоящим из количества слов до 10. Основополагающей составляющей данной проблемы является неоднозначность. Сарказм является крайне субъективным как для восприятия, так и для понимания. Данная задача ставилась пока только для английского языка. Определению сарказма способствовал анализ контекста, начиная от анализа соседних слов, проверкой на повторение слова, так и вычислением слов, которыми чаще всего используются для сарказма. Общий вид задачи выглядел так - давался набор тренировочных данных, в котором имелись следующие параметры: метка, является ли предложение сарказмом или нет; сообщение; автор сообщения; предыдущее сообщение; оценка других пользователей данного сообщения; время публикации; общая оценка.

Для решения такой задачи применялось множество методов машинного обучения и наиболее популярные, а более того эффективные, это: рекуррентные сети с краткосрочной и долгой памятью, тематическое моделирование и метод опорных векторов.

### 1.4 Задача распознавания символов

Проблематика распознавания символов заключается в том, что символ является также неоднозначным и порою субъективным параметром. Поэтому введем определение символа. Символом будет считаться слово или словосочетание, значение которого зависит от культурологического контекста. Символ часто пересекается со стилистическими приемам, особенно с:

* Аллитерация - повторение в стихотворной речи одинаковых согласных звуков с целью усиления выразительности
* Анафора - Единоначатие, повторение слова или группы слов в начала стихотворных строк
* Антитеза - Резкое противопоставление понятий
* Амплификация - Нагнетание в тексте синонимичных тропов или однородных конструкций
* Гипербола - Преувеличение действия, предмета или явления
* Ирония - Оборот в основе которого лежит контраст видимого и скрытого смысла высказывания
* Каламбур - Основывается на игре слов, их звуковом сходстве при различном смысле
* Метафора - Слово или выражение, употребляемое в переносном значении
* Олицетворение - присвоение свойств одушевленных предметов неодушевленным
* Перифраз - заключается в замене слова или словосочетания описательным оборотом речи, в котором указаны признаки неназванного прямо предмета
* Сравнение - в нем происходит уподобление одного предмета или явления другому

В разных типах литературных произведений выделение символа отличаются, как видно из этого списка. Также от автора к автору меняется как тематика, так и выделяющийся объект. Для больших произведений имеется смысл разделять текст на логические блоки. Примером служат романы, где от главы к главе меняется место действия, лица и время, что резко меняет направление повествования. В романе может параллельно вести две и более сюжетные линии, и на протяжении них выделяются различные символы. В стихотворениях особенность исследования заключается в ограниченности текста при неизвестном количестве символов в нем, которое может варьироваться от их отсутствия до наличия символов в каждом блоке.

Для анализа, обучения и настройки алгоритмов выбраны яркие представители русского символизма Александр Блок и Михаил Булкагов. В их произведениях отображалась жизнь каждой души, которая полна переживаний, смутных и неясных сомнений, тонких чувств, мимолетных впечатлений. Они создавали два мира - мир вещей и мир идей, при этом символ в их произведениях или ряде произведений становился неким условным знаком, который объединял эти два мира. Чаще всего символ связан с ирреальным миром, делая символику чем-то сакральным, священным и святым.

Таким образом выделяются черты символизма: индивидуализм; идеализм; преобладание общего над частным; образность; семантизм по отношению к возвышенным и низменным аспектам жизни, а также государственного строя; созерцательность.

Еще одной проблемой распознавания символов является вариативность, поскольку помимо анализа слова и предложения требуется анализировать словосочетания. Это приводит к росту требуемой памяти, времени обработки и скорости работы. Более того, так как символ может быть упомянутым на протяжении всего произведения, приходится держать в памяти весь текст и сопоставлять слова и словосочетания с ним.

## ПРАКТИЧЕСКАЯ ЧАСТЬ

### 2.1. Применяемые языки программирования, программные обеспечения, базы данных и инструментарий.

#### 2.1.1 Python

В качестве основного языка программирования для разработки был выбран язык Python. Python - активно развивающийся высокоуровневый язык программирования, отличающейся динамической типизацией. Широко распространен и адаптирован для работы на многих современных платформах.

Первичные модули, играющие ощутимую роль в предобработке данных - Nltk и Pymorphy2. Pymorphy2 - это комплексный морфологический анализатор, работающий со словарем OpenCorpora, строящий гипотезы для незнакомых слов и работающий со скоростью до 100 тысяч слов в секунду. Его особенности - приведение слов в нормальную форму, возможность поставить слово в нужное число или падеж, а также вывод грамматической информации о слове.

Nltk - модуль для синтаксического анализа предложений. Основной функционал - стемминг, лемматизация, избавление от малоинформативных слов и обработка знаков пунктуации. Nltk включает в себя различные алгоритмы для каждого функционала, что делает модуль гибким и удобным. Также для большинства операции имеет нейронную сеть.

Natasha - модуль для извлечения имен собственных для людей, животных, улиц, городов и различного вида прозвищ, а также извлечение времени и места для русского языка. Данный модуль работает на основе Pymorphy2 и парсера Yargy c использованием внутреннего словаря. Имеет несколько видов работы с именами. В упрощенной версии не учитывается контекст и не используется нейронная сеть, что делает работу значительно быстрее.

Модули doc, csv и pandas применяются для работы с документами типа docx и xls. Doc позволяет считывать, обрабатывать и записывать документы, при необходимости как сохраняя функционал Microsoft Word, так и минимизируя для удобства. Модули csv и pandas широко применяются в машинном обучении для считывания данных различного вида.

Модуль gensim содержит в себе множество базовых алгоритмов машинного обучения с возможностью настройки и модификации. Включает в себя функционал для работы с текстами, их предобработки и векторизации. Особую популярность составляют методы fasttext и Word2vec, которые помимо стандартного эмбединга слов согласно их встречаемости в тексте учитывают их контекстное значение, а также возможность применения в устойчивых выражениях в дополнительном скрытом слое матрицы.

Библиотека BigArtm предлагает широкий ассортимент моделей для непосредственной работы с текстом. Позволяет проводить тематическое моделирование больших коллекций документов, а также эффективную и потоковую параллельную реализацию вероятностного тематического моделирования на основе аддитивной регуляризации.

#### 2.1.2 Тематическое моделирование

Основным методом, применявшимся для исследований этой задачи, являлось тематическое моделирование. Его основная цель - это построить модель коллекции текстовых документов, которая определяет, к каким темам относятся документы и какие слова образуют каждую тему. Применение тематического моделирования хорошо описано в [3] и [8]. Данный метод имеет широкое применение в классификации, аннотации и поиске документов.

В терминах тематического моделирования общая постановка задачи выглядит следующим образом. Необходимо найти темы имеющие распределение и на множестве и , символы , где - документы во множестве документов, - множество слов из всех слов, встречающихся во всех документах.

Также в тематическом моделировании используется гипотеза условной независимости, которая утверждает, что распределение всех слов в документе и относящихся к определенной теме описывается общим уравнением распределения слов по темам и не зависит от самих документов. Это представлено тремя эквивалентными уравнениями:

Таким образом, исходя из данной гипотезы, определения условной вероятности и формуле полной вероятности мы получаем вероятностною модель порождения данных

В данной работе гипотеза об условной независимости приниматься не для всех алгоритмов по следующим причинам:

1. Символ состоит как из одного слова, так и из словосочетаний, что приводит к зависимости слов от друг друга в контексте
2. Описание символов в базе данных является общим, поэтому в таком случае условная зависимость упрощает выделение единственной темы и слов ее образующих. И лишь затем на основе слов, образующих тему символа, будет основываться один из аспектов выделения символа.
3. Тематика текста может включать в себя символ, в случае акцентирования его внимание в данном тексте.

В качестве основной модели для тематического моделирования была выбрана LDA, в которой векторы документов и векторы тем порождаются распределениями Дирихле. Причин для ее выбора было множество: простая настройка гиперпараметров и каждая из которых влияет на выделение тем и распределение слов в них. за различие документов между собой, а отвечает за различность тем. Повышение различности зависит от понижения и . У распределения Дирихле имеется особенность, заключающаяся в выделении редко используемых слов. Это важно для анализа выборки, построенной на текстах из художественной литературы.

Применение тематического моделирования к большим произведениям при малом количестве тем позволяет выделить его общую направленность, а применение на отдельных абзацах или главах - соответствующую направленность выбранной части текста.

#### 2.1.3 LSTM

В качестве показательного сравнения отдельно от проекта [20] используется нейросеть BiDirectional LSTM с использованием CRF. LSTM-модуль — это рекуррентный модуль сети, способный запоминать значения как на короткие, так и на длинные промежутки времени. Ключом к данной возможности является то, что LSTM-модуль не использует функцию активации внутри своих рекуррентных компонентов. Поэтому хранимое значение не размывается во времени, и градиент или штраф не исчезает.

Задача поиска символов в тексте может сводиться к sequence labeling – маркировке последовательности. Рассматривается два подхода к задаче: с точки зрения классификации и с точки зрения sequence labeling.

1. **Sequence labeling**: Дано предложение **X**, разделенное на части (слова): Требуется построить последовательность двоичных меток (labels) которые идентифицируют наличие неоднозначности в каждом слове
2. **Классификация:** Аналогично дано предложение **X,** разделенное на части (слова): . Требуется для целевой переменной предсказать отношение к классу символ или не символ, соответственно 1 и 0.

В целом, Sequence labeling является обобщением классификации при такой постановке, поэтому задачу можно описать так:

множество слов предложения, -- множество ответов (отношение к классу 1 или классу 0). Требуется построить алгоритм способный классифицировать произвольный объект

Модель получает на вход последовательность слов и предсказывает для каждого из них соответствующую ему метку – символ/не символ. Для начала слова переводятся в векторное пространство, в результате чего получается последовательность векторов из этого пространства Далее, эти векторные представления подаются на вход двум LSTM компонентам, двигаясь по тексту в различных направлениях, таким образом создавая представления для конкретного контекста. Соответствующие прямые и обратные представления конкатенируются для каждого положения слова:

Затем, добавляется скрытый слой нелинейности:

где весовая матрица между слоями.

В конце для создания самих меток используется либо softmax, либо Conditional Random Fields, разновидность метода Марковских случайных полей, в зависимости от выбранной постановки задачи. Функция softmax рассчитывает нормированное распределение вероятностей по всем возможным меткам для каждого слова:

где вероятность того, что метка -ого слова будет ( множество всевозможных меток), k-ая строка весовой матрицы

Для оптимизации модели используется минимизация отрицательного логарифма вероятности верной метки:

Таким образом решается задача оптимизации:

В итоге мы получаем модель, способную верно классифицировать символы при сильно отличающихся данных.

#### 2.1.3 Базы данных и платформы для работы с ними

В качестве вспомогательной системы реализована информационная база символов. Первообразом данной базы является энциклопедия из [18], которая имеет своей задачей сбор широкого спектра сведений о символике и хранит символы с различными интерпретациями. Данные толкования включают в себя религиозные, культурологические и исторические способы объяснения символа. Такая структура информации о символах позволяет активно применять тематическое моделирование для каждого из них, однозначно определяя темы и слова, входящие в темы.

Обобщенность символов имеет множество плюсов и помогает выделять достаточно большое количество символов, если они присутствуют в произведении. Тем не менее у каждого автора существует свои стилистика, влияющая на вид произведения, тип повествования и способ встраивать символы в текст. Кроме того, у некоторых писателей существуют сборники, объединенные по единому архетипу и сопровождающие одну тематику. В таком случае требуется использовать более точно настроенные алгоритмы и учитывать символику, определенную на других схожих произведениях, так как точность будет напрямую зависеть от этого.

Централизованного хранения размеченных символов достаточного количества для определенных произведений на текущий момент не существует, что является существенной проблемой. В литературе имеются научные работы, связанные с исследованиями письменных романов символистов, но доступ к ним в большей части закрыт. Для обучения, настройки и проверки необходимо большое количество исследуемых данных, разделенных по типу произведений и авторам. Более того, разметка должна быть произведена экспертами.

Для должного и оперативного взаимодействия словарем в качестве динамической платформы разработан сайт, представляющий из себя электронную библиотеку. В сайт интегрирован механизм, реализующий алгоритм распознавания символов. В целях оптимизации работы и удобства взаимодействия пользователя с функционалом символы разделены на предполагаемые или на постоянные, которые подтверждены эксперты. Возможность просмотра символов доступна при ознакомлении с литературным произведением и в качестве отдельного списка символов, с различным толкованием и ссылками на произведения, где оно используется. Общий размер словаря символов составляет 7758 символов.

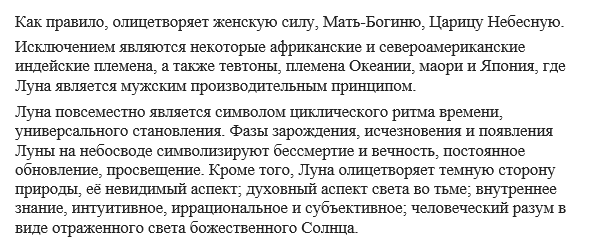


Рис. 2.1 Описание символа

Основная структурная модель сайта написана на Django, а для взаимодействия с базой данной SQL используется СУБД Postgresql. Такое решение было принято для минимизации проблем, возникающих с работой большого количества произведений и большего количества символов. Особенности Postgresql позволяют хранить неограниченный размер базы данных с максимальным размером таблицы равным 32 Тбайта и максимальным размером поля в 1 Гбайт. Он не ограничен лимитом в количестве записей в таблице, в то время как таблица вмещает в себя до 1600 полей.

Концепт взаимодействия всей базы данных на сайте:

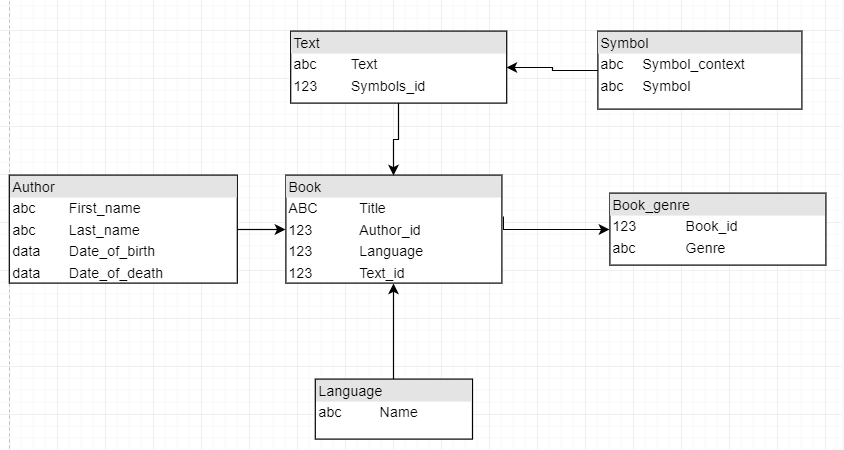


Рис. 2.1 База данных сайта

### Предобработка данных.

Далее рассмотрим схему работы алгоритма, представленном на рисунке 2.2. Работа алгоритма составлена при помощи стандарт IDEF0, что позволило интегрировать данный алгоритм для решения различных задач в ходе исследования без необходимости изменения каждого отдельного блока, и значительно ускорило отладку и настройку каждого параметра. На вход может подаваться обрабатываемый текст в виде файла формата .txt, .doc и единой строкой через сайт.

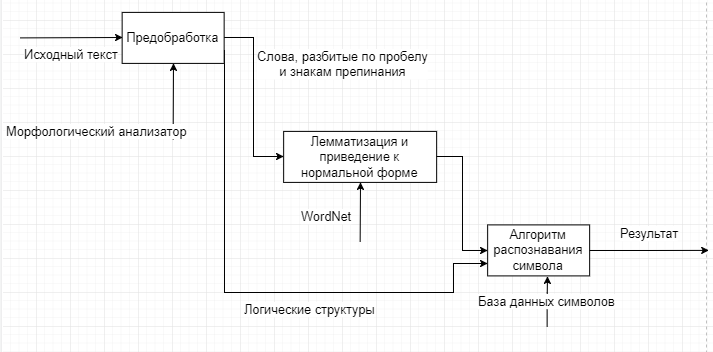


Рис. 2.2 Блок-схема работы алгоритма согласно IDEF0

Алгоритм предобработки принимает в качестве единственного аргумента строку, представляющую весь текст или его часть. Уже внутри ее работы вызываются функции разбиения на слова и лемматизации. Функция разбиения на слова принимает весь текст, после чего производит разбиение по предложениям и на слова внутри каждого предложения, избавляется от ненужных знаков и малоинформативных слов. На выходе функции в качестве результата возвращается массив из списка слов. Аргументом в функции лемматизации выступает слово, которое проходит лемматизацию. Если слово является редко используемым, именованием или распространено лишь среди определенного круга лиц, то слово не приводится к нормальной форме и возвращается в исходном виде. Иначе же на выход получаем преобразованное слово. В результате работы исходной функции предобработки получаем массив из списков слов. Каждый список слов представляет собой предложение, разбитое по словам и сохраняющий порядок последовательности. Текст может предоставляться независимо от источника, главным условием является лишь наличие стандарта кодирования utf-8.

Результат работы данного блока позволяет выделить базовую логическую единицу - слово. Нивелирование наличия союзов, предлогов и междометий значительно улучшает определение значения слова, так как в первоначальном тексте количество данных частей речи составляет до от всех слов и негативно влияет на частотность встречаемости слов в предложениях.

Перед разбиением на словесные структуры идет извлечение имен, именованных сущностей и названий местоположений. Для каждого человека или существа выделяется его имя, фамилия, отчество или второе имя, а также кличка или позывной. Возможно выделить отдельно должность человека. После выделения всех сущностей в отдельный массив, следует минимизация количества элементов в нем путем поиска одинаковых упоминаний в тексте. Затем систематически происходит поиск упоминания объекта с использованием лишь части его полного имени на основе логического повествования, чтобы еще более минимизировать размер массива. Выделение именованных сущностей является необходимым, так как согласно логике, они выполняют роль существительных в предложении и не включены в корпуса русского языка.

И только потом происходит разбиение предложения на большие структуры, чем одиночные слова. Аксиоматика конструирования таких структур представлена следующими правилами:

1. В качестве основы берется слово, часть речи которого является существительное, глагол, числительное или имя собственное.
2. Предикативная связь между членами грамматической основы представлена тремя связями:

* Согласование - вид связи, при котором зависимое слово согласуется с главным в роде, числе, падеже. Главное слово всегда существительное; зависимое может являться: прилагательным, причастием, местоимением или числительным.
* Примыкание — вид связи, при котором зависимость слова выражается лексически, порядком слов и интонацией, без применения служебных слов или морфологического изменения. Образуется наречиями, инфинитивами и деепричастиями, а также притяжательными местоимениями 3 лица, простой формой сравнительной степени прилагательного или наречия.
* Управление - вид подчинительной связи, где зависимое слово находится в форме косвенного падежа.

1. Тип выстраиваемой связи напрямую зависит от последовательности обработки предложения
2. Отдельно выделяется связка подлежащего и сказуемого, поскольку они несут основную информативную часть предложения.

Результат работы данного блока представляет собой воссоздания словесной структуры и множества .

После того, как текст прошел предобработку и был разбит на слова, необходимо привести одни и те же слова в разной форме в единую. Но прежде чем слово начнет проходить обработку, специально для русского языка происходит систематизированное приведение буквы Ё к Е, чтобы минимизировать вариации написания слов.

Затем слово проходит по внутреннему словарю из 400 тысяч слов с различными вариациями склонений. В случае, если из одного исходного слова исходит множество слов в нормальной форме, выстраивается вероятностная догадка по его происхождению на основе всего предложения. В случае слова Белом предполагается слова в нормальной форме Белый и Бел. Дальше идет семантический и морфологический анализ этих слов, проверяется совстречаемость данных слов с другими и проверка на структуру предложения со стороны грамматики русского языка.

На случай обработки редко используемых или специфичных слов больше чем 5 букв выстраивается следующая система правил:

1. Проверяется наличие префикса в слове среди известных префиксов. В случае, если слово начинается с него, то он отсекается, после чего оставшаяся часть анализируется и префикс в конце приписывается обратно
2. Подбирается длина префикса: в начале одна буква, потом первые две буквы, потом первые три буквы и так до пяти, и каждый раз проверяется наличие остатка в словаре.
3. Аналогичным способом идет проверка окончаний, за исключением того, что самые частые окончания не хранятся в памяти, а вместо этого выполняется проверка корней известных слов.

В итоге данного блока формируется окончательный вид слова и множества .

### 2.3 Постановка задачи

Пусть имеется множество документов или текстов . Каждое содержит в себе множество слов , которые и составляют документ согласно некоторому распределению . Каждое слово в свою очередь имеет одно и более смысловое значение .

Определение 2.1. Смысловым значением будем называть интерпретируемость слова относительно контекста.

Смысловое значение может как иметь зависимость определения напрямую от контекста, в случае если слово имеет множество интерпретаций, так не требовать проверки контекста в случае, если множество определений имеет лишь единственный элемент. Контекст определяется по словам в том же предложении и по ближайшим соседним предложениям, так как применяемые в одинаковой тематике и значении слова окружают схожие слова и предложения. Стоит учесть, что разные слова могут обладать одинаковыми смысловыми значениями. В данном исследовании именно смысловое значение является базисной единицей.

Исходя из алгоритма предобработки и лемматизации мы получаем необходимые множества слов , текстов и . Также в ходе работы мы получаем множество словесных структур .

Определение 2.2. Словесной структурой в тексте будем называть множество из слов , распределенных в строгой заранее заданной последовательности и согласно известному распределению , а равно количеству всех слов в документе, и являющихся неким подмножеством множества .

Во множестве, образующим P, слова W также соблюдают данную последовательность, являясь соседями в . Соседями будут считаться элементы - количество соседей.

Тогда мы получаем следующие выводы:

* Во множестве каждый элемент является подмножеством в
* Множество не является равным множеству .

Основной задачей является поиск символов и всего множества символов , при имеющихся , множества слов , распределения . Исходя из ранее данного определения символа в 1.4 получаем, что каждый символ имеет распределение по множествам и . Каждый символ может строго входить либо во множество , либо во множество . Благодаря этому задача может быть сведена к задаче поиска символов S на основе распределений и

Определение 2.3. Слово будет считаться символом , если выполняется условие

где - количество слов в словесной структуре, в которое входит слово, – итерационный весовой параметр. Функция возвращает 1 или 0, в зависимости от того, входит ли проверяемое слово в тематику любого символа из базы данных.

Определение 2.4. Словесную структуру будем определять как символ, если выполняется определение 2.3 для любого из слов , входящее во множество слов словесной структуры .

Выделим несколько гипотез относительно природы символа, для улучшения анализа и поиска символов:

* Символ имеет вероятностное распределение среди документов
* Символ обладает зависимостью от слов в тексте
* Символ НЕ обладает зависимостью от очередности слов в тексте.
* Символ совпадает с темой текста.
* Для определения символа необходима обучающая выборка, созданная на основе экспертной разметки

На этапе обучения и настройки задача поиска символов представляет собой задачу, когда при имеющихся множествах символов , документов , слов и распределения слов по документам построить зависимость и .

Размеченные ранее символы выполняют функцию базы данных символов, а также образуют общее множество символов S. На ее основе и строятся вероятностные распределения символов. Символ представлен в базе как слово или словосочетание в качестве основного ключа и один или более описание значения символа. Сюда же входят предложения из художественных произведений, в которых символ был выделен, определен и подтвержден экспертом. На основе каждого описания выделяется тема и слова , образующие ее. Наличие большого количества произведений и авторов приводит к необходимости воссоздания зависимостей и специфичных только для определеного автора. Тогда хранение общего распределения и распределения, вычисленного для каждого конкретного автора, приводит к росту вычислительных операций и хранимой памяти, но при этом растет и точность определения символа. В итоге общая накопительная информационная система представляет усредненное распределение символов, которая по факту позволяет определять наиболее обобщенные по смыслу символы. А каждого из авторов является возможным рассчитать отдельное вероятностное распределение согласно их стилистике написания.

### Описание экспериментов.

В течении всей исследовательской работы проводились серии экспериментов на основе различных моделей, правил обработок и входных данных.

#### 2.3.1 Эксперимент 1.

Первый эксперимент состоит из 2 этапов: работа с поэзией и прозой. На первом этапе в качестве основы работы предполагалось использование тематического моделирования для текста с целью выделить в качестве темы символы. Обрабатываемые тексты представляли собой стихотворения Александра Блока. Они обладают следующими особенностями:

* небольшой размер текстов, что позволяло производить быструю обработку;
* наличие большого количества многозначных конструкций;
* неявная фиксация на определенном тематическом направлении;
* экспрессивность выражений;
* сильная контекстная зависимость слов.

Изначально работа велась только со стихотворениями при помощи LDA без использования внешних баз данных. В ходе исследования обнаружилось, что выделить структурно символы в стихотворениях является плохо решаемой задачей. Этому способствовали малый размер текста, малое повторение слов и структурные особенности произведений. Отсутствие корпусной составляющие и системы правил при работе со стихотворениями приводило к тому, что каждое слово было уникально, взаимосвязь слов по смыслу между собой не учитывалась, а первоначальная гипотеза мешка слов лишь ухудшало алгоритм.

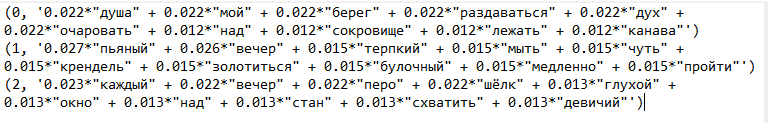


Рис. 2.3 Результат работы LDA для

На втором этапе тексты были представлены романами Булгакова. Результат стал значительно лучше, так как в таком случае тематическое моделирование выделяло в словах, образующих тему символы, главным образом те, которые явно определяются в ходе всего произведения. При работе с большим текстом было выявлено, что размерность текста сильно влияет на работу. С ростом количества элементов множества слов, составляющий обрабатываемый текст, росло качество разбиения на темы до определенного порога. Чрезмерная размерность в последствии может ухудшить пороговое значение, но незначительно. Поэтому разбиение текста для последующего анализа ведет к улучшению его обработки. В качестве постоянного разбиения был выбран анализ по главам, так как они содержат строгое количество персонажей, участвующих в сюжете и посвящены определенному отрезку времени, в ходе которого практически нет разрыва в действиях. Работа с текстом показала важность предобработки текста перед непосредственной обработкой. Более того, из-за логики работы тематического моделирования требовалось избавиться от уникальности наименований и структурно оптимизировать подаваемые данные, как было сказано ранее.

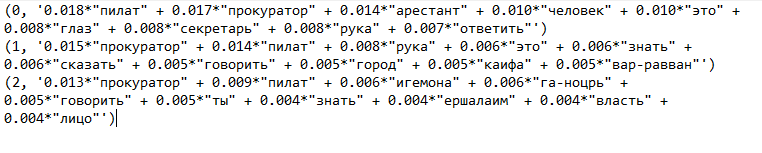


Рис. 2.4 Результат LDA для главы 2

Так как в случае тематического моделирования на основе текста символы выделялись согласно вероятностному распределению, то предположение о наличие символа строилось случайным образом и не имело конструктивной базы. Также слова, образующие тему, могли не иметь связности, а их выделение приводило к замешательству.

#### 2.3.2 Эксперимент 2

Выводы, сделанные на основе первого эксперимента, привели к необходимости выделения символа на основе некоторого правила. Поэтому следующим экспериментом стало применение словаря символов. В таком случае большое значение имело вхождение слова и словосочетаний в базу данных. Этот способ выделения символа был строгим и соответственно наивным, так как применение не затрагивало включение контекста используемых символов и это давало большое количество неверных определений. Тем не менее само выделение символов в произведениях стало более простым, явным и постоянным. Количество выделяемых слов при повторной работе было константным, как и количество верно выделенных символов. По сравнению с первым экспериментом среднее количество выделяемых символов возросло. Номинальная точность алгоритма стала высокой, но вот метрика качества классификации оставалась в районе погрешности.

#### 2.3.3 Эксперимент 3

Целью нового эксперимента был поиск способа уменьшить количество выделяемых слов алгоритмом в качестве символа, но при этом не являются символами со стороны экспертной разметки. За основную гипотезу был взят тот факт, что символом чаще всего нужно считать словосочетание и дальнейшая работа была направлено в этом направлении. Данный эксперимент состоит из 2 этапов. В первом предполагалось анализировать данные на основе идеи N-gramm, где бралось слово и окружающие его слов. В качестве плюсов предполагалась гибкость настройки количества анализируемых слов, настройка весового параметра, отвечающего за принятие предположений. По мере работы из расчетов оказалось, что работа алгоритма стала требовать значительно больше времени, линейно зависящее от параметра .

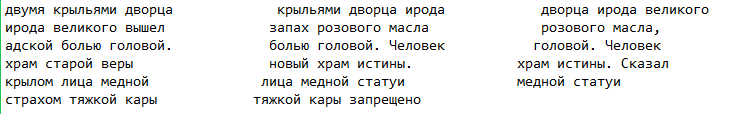


Рис. 2.5 Результат обработки N-Gramm

На втором этапе альтернативой применению N-Gramm стало выделение словосочетаний на основе правил сочетания слов согласно правилам русского языка. Таким образом нивелировалась необходимость в реверсивной предобработке данных, и работа алгоритма относительно текста стала последовательной. Внедрение такого типа выделения данных позволило сократить ошибочность выделения данных более чем в два раза.

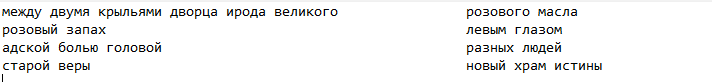


Рис. 2.6 Результат обработки словосочетания

#### 2.3.4 Эксперимент 4

Целью данного эксперимента являлось увеличение информационной базы символов при помощи дополнительных внешних источников данных. В эксперименте представлено два этапа.

Первый этап рассматривает расширение при помощи специализированных сборников тематического толкования символов. Сборники включают в себя толкование при строго определенном контексте образов, которые возникают у человека в подсознании. Образ, место действия и происходящее может не иметь взаимосвязи в описываемых событиях. При этом существует строго определенное трактовка образа, которое никак не связано напрямую с элементами события. Различные попытки использовать данный набор данных не привели к успехам. При слишком высоком пороге количество выделяемых символов быстро стремилось к нулю, так как требовалось соблюдения всех параметров, которые в художественной литературе не встречаются из-за отсутствия взаимосвязи. При незначительном уменьшении порога наблюдался ровно противоположный эффект и точность классификации была нулевой.

Второй этап включал в себя размеченный корпус русского языка. Корпус русского языка представляет собой огромный толковый словарь. Преимущество словаря [19] состоит в том, что в нем слова, которые в определенном контексте имеют переносный смысл, помечаются отдельно с сохранением предложения содержащим его. Однако, так как пополнение корпуса производилось автоматизировано, то многие данные повторялись или оказались шумом. Поэтому без ручной разметки дальнейшее развитие оказалось невозможным.

Тем не менее применение в качестве дополнительного модификатора к имеющейся базе символов позволяет улучшить результативность. Основное решение происходит при анализе на основе первоначальной базы символов, а совпадение или сопоставление результатов на одном промежутке текста позволяет внести аддитивный вес на окончательное решение. Таким образом, незначительно расширяется или уменьшается выходная выборка, в зависимости от используемой дополнительной базы.

Таблица 2.1 Сравнительные характеристики двух баз данных

|  |  |  |
| --- | --- | --- |
| Информационная база | Преимущества | Недостатки |
| Специализированный сборник | При высоком пороге и крайне тонкой настройке позволяет единично выделять символы в тексте | Не годится для поиска всех символов в тексте, имеет ограниченность использования из-за отсутствия вариативности описаний |
| Корпус русского языка | Позволяет дополнительно оценивать предложенные символы на многозначность, исходя из самих слов | Недостаточность ручной разметки и повторяемость контекстов приводит к невозможности использования в автономном режиме |

#### 2.3.5 Эксперимент 5

Следующий эксперимент был направлен на применение тематического моделирования. Таким образом дальнейшим этапом разработки ПО стало следующее - тематическая модель применялась для описания символов в исходной информационной базе символов, а также контекста предложения в уже найденных отрезках художественных произведений. При такой работе возникли проблемы. Во-первых, было неизвестно заранее количество описаний при автоматизированной работе. Во-вторых, системе было необходимо держать в памяти большое количество информации. Но даже на небольшом количестве обработанных символов эта идеология позволяла выделять символы в произведении на основе схожести анализируемых предложений с одним из имеющихся толкований символа, что несомненно повысило точность и надежность алгоритма.



Рис. 2.7 Анализ словосочетания с LDA для базы символов

Чтобы удовлетворить потребность в хранении большое количество данных, было решено разработать динамическую платформу, способное это сделать. Выбор пал на сайт, а в качестве языка разработки – Django. Основной причиной стала совместимость данного языка с Python. Главным функционалом для сайта представляется структурное хранение таких данных, как авторы, произведения и символы. Также подразумевается привлечение большого количества экспертов для разметки и улучшения обработки текстов. Таким образом возможно внедрение алгоритма для создания динамического помощника, который будет выдавать предположения о символе для дальнейшего подтверждения. Окончательным мнением будет являться решение эксперта, после чего верно размеченные символы пополняют базу данных вместе с предложением, в котором оно располагается.

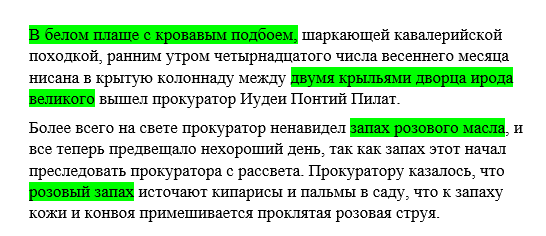


Рис. 2.8 Результат работы алгоритма с выводом в docx

#### 2.3.6 Проверка качества моделей.

Самая простая оценка точности классификации алгоритма на основе соотношения правильных ответов к общему количеству не является приемлемой, так как количество слов, словоформ и словосочетаний много больше чем символов, что приводит к непропорциональности изначального соотношения. Такая метрика присваивает всем документам одинаковый вес, что может быть не корректно в случае если распределение документов в обучающей выборке сильно смещено в сторону какого-то одного или нескольких классов. Таким образом точность может достигать высоких значений, но практическая эффективность будет стремится к нулю. Поэтому для такого анализа введем точность и полноту для оценки точности.



Рис. 2.9 Таблица контингентности

Точность (precision) и полнота (recall) являются метриками которые используются при оценке большей части алгоритмов извлечения информации. Иногда они используются сами по себе, иногда в качестве базиса для производных метрик, таких как F-мера. Суть точности и полноты очень проста.

Точность системы в пределах класса – это доля документов, действительно принадлежащих данному классу относительно всех документов, которые система отнесла к этому классу. Полнота системы – это доля найденных классификатором документов, принадлежащих классу относительно всех документов этого класса в тестовой выборке.

Согласно рисунку 2.3 мы классифицируем ответы алгоритма с ответами экспертов, а именно:

* **TP** – Истинно-положительный ответ
* **TN** – Истинно-отрицательный ответ
* **FP** – Ложно-положительный ответ
* **FN** – Ложно-отрицательный ответ

Из этого точность и полнота определяются следующим способом:

Данные метрики мы будем использовать для расчета окончательной метрики, которая будет представлять собой среднее гармоническое между точностью и полнотой.

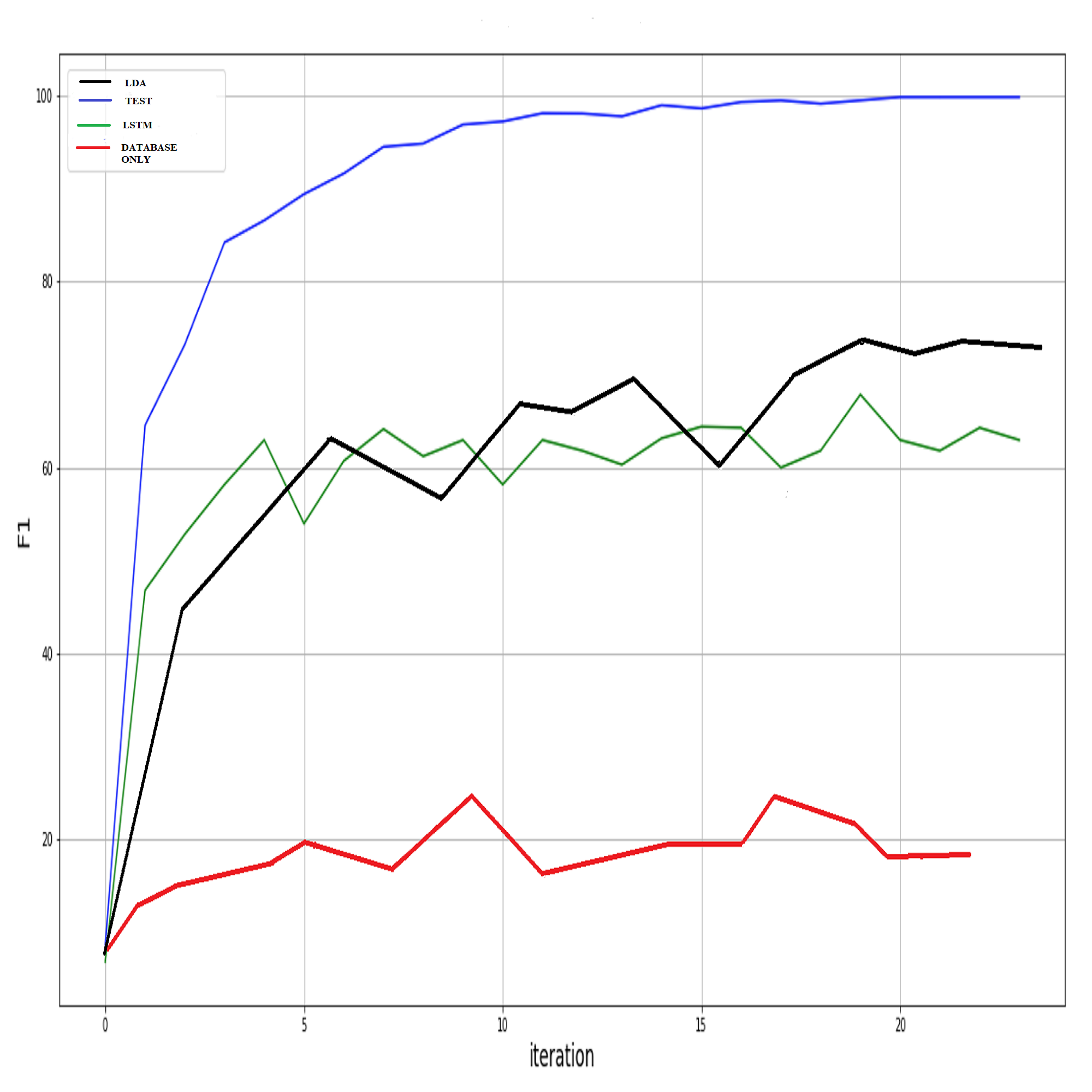


Рис. 2.10 Графики сравнения алгоритмов на 10 главах

Таблица 2.2. Таблица сравнения результатов различных моделей

|  |  |
| --- | --- |
| Алгоритм | F-мера |
| **LDA** | 0.049 |
| **Symbolarium** | 0.193 |
| **LDA + Symbolarium** | 0.661 |
| **LSTM** | 0.658 |

Проверка качества классификации проводилась на отдельной выборке, размеченной экспертами. Данная выборка состоит из 10 глав произведения . Так как данные подвергались очистке, чтобы гарантировать их качество, то они подходят для метрики.

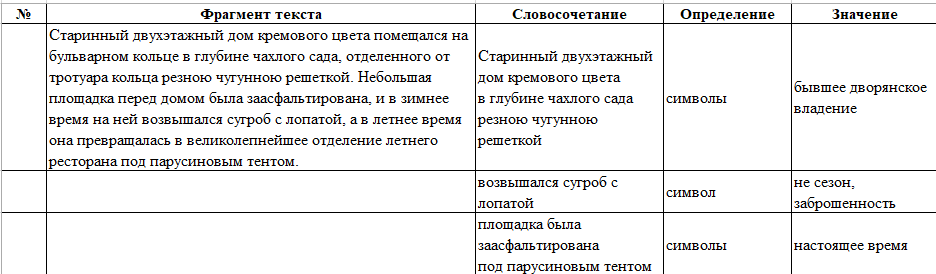


Рис. 2.11 Пример экспертной разметки

Данная метрика применялась для всех алгоритмов, кроме первоначального, поскольку его эффективность, согласно метрике, стремилась к нулю. На рисунке 2.4 рассматривается применение 3 методов на основе новых данных и качество на обучающей выборке, отмеченная ломанной линией синего цвета. Красным цветом показана эффективность метода, основанная на совпадении слова с наличием его в информационной базе символов. Два пересекающихся графика представляют собой сравнительный анализ двух моделей, разработанных под распознавание символов. Как видно на графике, тематическая модель имеет тенденцию к повышению точности определения символа на большем промежутке, но сильно завязано на обширности используемой базы данных. Такая модель имеет незначительное повышение качества относительно модели LSTM, которая достаточно хорошо работает даже на малом количестве итерации.

# ЗАКЛЮЧЕНИЕ

В ходе данного исследования был проведен комплексный анализ статей, связанных с машинным обучением и обработкой естественных языков. С каждым проведенным экспериментом расширялся инструментарий и повышалась точность распознавания. Рассматривались различные гипотезы и принципы взаимодействия компонент и благодаря этому подобран наиболее эффективный алгоритмический комплекс для решения задачи. Первоначальные гипотезы о символе были опровергнуты, однако дальнейшие исследования позволили выдвинуть новые гипотезы, которые были доказаны в ходе работы. Символ имеет сильное значение от контекста, в котором используется и зависит от ближайших слов, а также может состоять как из одного слова, так и словосочетания. Для сравнительного анализа применялась F-мера для LSTM, LDA, разработанных специально для проверки выдвигаемых гипотез.

В работе рассматривалось применение различных информационных баз по отдельности и совместно. Это позволило выделить преимущества и недостатки каждой базы, а также найти способ улучшения имеющих алгоритмов.

Дальнейшее пополнение базы данных значительно улучшит результаты. Уже сейчас программный комплекс позволит упростить экспертам поиски символов. Данное исследование показало, что при работе с текстом необходимо учитывать, как контекстный смысл, так и множество скрытых факторов, которые специфичны для определенного типа задач.

Перспективой внедрения наработок по данной теме является то, что открывается совершенно новый уровень абстракций при работе с текстом, и в возможности использования такого уровня абстракций в других направлениях машинного обучения.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Андреас Мюллер, Сара Гвидо, Введение в машинное обучение с помощью Python, 2011, С. 393
2. Батура Т. В. Методы автоматической классификации текстов //Программные продукты и системы. – 2017. – Т. 30.
3. Воронцов К. В. Вероятностное тематическое моделирование //Москва. – 2013, С. 80
4. Заботкина В.И Методы когнитивного анализа семантики слова: компьютерно-корпусный подход, 2014, 30
5. Корпус русского языка. URL: <http://www.ruscorpora.ru/old/> (дата обращения: 11.05.2019).
6. Платформа Kaggle – Открытый конкурс по распознаванию сарказма. URL: <https://www.kaggle.com/c/uncch-inls690-sarcasm-detection> (дата обращения: 11.01.2019).
7. Селегей В.П, Беликов В.И, Богуславский М.И Компьютерная лингвистика и интеллектуальные технологии, 2015, 774
8. Blei D. M. et al. A correlated topic model of science //The Annals of Applied Statistics. – 2007. – Т. 1. – №. 1. – С. 17-35.
9. David W. Aha, Kibler D., Marc K. Albert, Instance-Based Learning Algorithms, 1991, C. 37-66
10. Koltsova O. Y., Alexeeva S., Kolcov S. An opinion word lexicon and a training dataset for russian sentiment analysis of social media //Computational Linguistics and Intellectual Technologies: Materials of DIALOGUE 2016 (Moscow). – 2016. – С. 277-287.
11. LDA: A supervised topic model for credit attribution in multi-labeled corpora //Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1. – Association for Computational Linguistics, 2009. – С. 248-256.
12. Mohammad S. M., Turney P. D. Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon //Proceedings of the NAACL HLT 2010 workshop on computational approaches to analysis and generation of emotion in text. – Association for Computational Linguistics, 2010. – С. 26-34.
13. Mueller J., Thyagarajan A. Siamese recurrent architectures for learning sentence similarity //Thirtieth AAAI Conference on Ar-tificial Intelligence. – 2016, С. 210-211
14. Ovesdotter C., Roth D., Sproat R., Emotions from text: machine learning for text-based emotion prediction, 2005, C. 579-586
15. Rayson P., From key words to key semantic domains, 2008, C. 519-549
16. Snoek J., Larochelle H., Ryan P. Adams, Practical Bayesian Optimization of Machine Learning Algorithms, 2012, C. 2951-2959
17. Scott M., PC ANALYSIS OF KEY WORDS - AND KEY KEY WORDS, 1997, C. 233-245
18. Symbolarium – Открытая база данных символики URL: <http://symbolarium.ru> (дата обращения: 13.11.2018).
19. Thorstem J., Text Categorization with Support Vector Machines: Learning with Many Relevant Features, 1998, C. 137-142
20. Severilov P., LSTM для распознавания символов. URL: <https://github.com/Intelligent-Systems-Phystech/2019-Project-46/tree/master/Severilov2019SymbolsInTexts> (Дата обращения:05.06.2019)