

## РЕФЕРАТ

Магистерская диссертация содержит 00 страниц, 00 рисунков, 00 таблиц. Список использованных источников содержит 00 позиций.

РАНЖИРОВАНИЕ, РЕКОМЕНДАЦИЯ, ЗАПРОС, ПОИСКОВАЯ СИСТЕМА, СЕМАНТИЧЕСКАЯ БЛИЗОСТЬ ТЕКСТОВ, ОЦЕНКА КАЧЕСТВА РАНЖИРОВАНИЯ, МАШИННОЕ ОБУЧЕНИЕ, ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ, ЯЗЫКОВАЯ МОДЕЛЬ, BERT.

Выпускная квалификационная работа посвящена исследованию возможных подходов к решению задачи рекомендаций документов, семантически схожих с другим документом или с описанной в запросе ситуацией, а также проблемам оценки качества рекомендаций в условиях отсутствия разметки данных.

Во введении описывается поставленная задача, обосновывается актуальность выпускной квалификационной работы, рассматривается теоретическая и практическая значимость, определяются объекты, предмет, цель и задачи исследования, указываются методы решения проблемы, дается краткий обзор информационной базы знаний.

В основной части приводится описание объектов исследования, теоретически обосновываются подходы к оценке работоспособности модели рекомендаций в условиях отсутствия разметки, предлагаются новые функционалы качества для оценки и сравнения моделей рекомендаций, описываются модели рекомендаций. В разделе 1.2 приводится описание подхода к оценке работоспособности системы рекомендаций на основе оценок пользователей, формулируются гипотезы о качестве работы системы для различных групп запросов. В разделе 1.3 предлагаются новые методы оценки качества рекомендаций в условиях отсутствия разметки данных для задачи

ранжирования, приводится характеристика предлагаемых функционалов качества. В разделе 1.4 описываются архитектуры моделей рекомендаций, используемых для решения задачи ранжирования. В разделе 1.5 приводится описание процесса обработки текстовых данных для их дальнейшего использования в моделях рекомендаций. ... В разделе 1.7 описываются параметры задачи. В разделе 2.1 изложен алгоритм решения сформулированной задачи. В разделе 2.2 приводятся использованные в численном эксперименте параметры задачи. В разделе 2.3 описывается практическая составляющая работы, представлены результаты численного эксперимента.

Результатами выпускной квалификационной работы являются предложенный метод оценки работоспособности системы рекомендаций по оценкам пользователей, предложенные функционалы качества для оценки рекомендаций в условиях отсутствия разметки данных для задачи ранжирования, спроектированные модели рекомендаций документов по запросу, программная реализация алгоритмов рекомендаций, а также проведенный численный эксперимент.

В заключении подводятся итоги данного исследования и формулируются выводы по проделанной работе.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ. . . . .	5
ОСНОВНАЯ ЧАСТЬ. . . . .	8
1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ . . . . .	9
1.1 Описание объектов исследования . . . . .	9
1.1.1 Система рекомендаций . . . . .	9
1.1.2 Запрос . . . . .	10
1.1.3 Рекомендация . . . . .	11
1.2 Оценка работоспособности системы . . . . .	12
1.3 Функционалы качества рекомендаций . . . . .	14
1.4 Рекомендательные модели . . . . .	17
1.4.1 Лингвистические алгоритмы . . . . .	18
1.4.2 Классификаторы соответствия документа запросу . .	20
1.4.3 Рекомендации по векторной близости документа за- просу . . . . .	23
1.5 Предобработка текстовых данных . . . . .	24
1.6 Постановка задачи оценки вероятности обслуживания пре- следуемого игрока . . . . .	26
1.7 Описание параметров задачи . . . . .	28
2 ПРАКТИЧЕСКАЯ ЧАСТЬ. . . . .	30
2.1 Алгоритм решения задачи . . . . .	30
2.2 Параметры задачи . . . . .	33
2.3 Результаты численного эксперимента . . . . .	35
ЗАКЛЮЧЕНИЕ . . . . .	38
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ . . . . .	39
ПРИЛОЖЕНИЯ . . . . .	42

## ВВЕДЕНИЕ

В данной выпускной квалификационной работе рассматривается система рекомендаций документов, которые были бы семантически близки запросу пользователя. В данной системе по запросу пользователя должен быть получен список ранжированных документов, где первый документ является наиболее близким запросу по смыслу, а каждый последующий имеет меньшую семантическую близость с запросом, чем предыдущий. При этом запросы в системе отличаются юридической спецификой и являются описанием какого-либо правового случая.

В данной работе решается проблема оценки работоспособности системы рекомендаций на основе анализа поведения пользователей, в условиях отсутствия разметки данных предлагаются новые функционалы качества для оценки эффективности алгоритмов рекомендаций, а также сравниваются различные подходы к построению алгоритмов рекомендаций.

Результаты данной работы будут использованы в компьютерной справочной правовой системе. Потенциальным приложением результатов работы является модификация алгоритма рекомендаций семантически схожих документов в реальной поисковой системе.

Появление данной задачи обусловлено необходимостью понять, эффективен ли существующий подход к построению рекомендаций, насколько эффективен, может ли быть улучшено качество рекомендаций и какими подходами и методами. Такие вопросы возникают ввиду растущей потребности пользователей реальной справочно-правовой системы в поиске документов, подходящих под описываемую ситуацию или под конкретный правовой акт.

Объектами исследования являются система рекомендаций в целом, а

также непосредственно запросы и рекомендации.

Предметом исследования выступает подход к построению рекомендаций и оценке их качества в условиях отсутствия разметки данных.

Цель данной работы - разработка методов оценки качества рекомендаций в условиях отсутствия разметки данных, построение алгоритмов рекомендаций семантически близких запросу документов, а также получение оценок качества разработанных алгоритмов с помощью предложенных функционалов качества.

Основными задачами **выпускной квалификационной** работы являются:

1. Поиск и обработка информации по объектам исследования;
2. Оценка качества существующего подхода к построению рекомендаций при использовании информации о поведении пользователей в поисковой системе;
3. Разработка функционалов качества рекомендаций в условиях отсутствия разметки данных;
4. Выбор и анализ подходов к построению моделей рекомендаций семантически схожих документов;
5. Обработка и подготовка текстовых данных для проектирования и обучения моделей рекомендаций;
6. Построение и обучение моделей рекомендаций;
7. Оценка качества построенных моделей с помощью предложенных функционалов качества;
8. Анализ полученных результатов.

По итогам выполнения данной **выпускной квалификационной работы**

поставленные задачи были успешно решены. Результат подтверждает компетентность предлагаемых методов оценки качества рекомендаций в условиях отсутствия разметки, а также указывает на конкурентоспособность разработанных алгоритмов рекомендаций по сравнению с текущим решением.

Данная работа развивает описанные в [1] идеи по построению ранжированных рекомендаций в поисковых информационных системах при комбинировании методов векторизации текстовых признаков, описанных в [2, 3, 4], с предиктивными моделями [4, 5, 6, 7]. Также данная работа предлагает новые подходы к оценке качества получаемых в условиях отсутствия разметки рекомендаций. В основе предлагаемых подходов лежит теория, описанная в [8, 9].

## ОСНОВНАЯ ЧАСТЬ


## 1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

### 1.1 Описание объектов исследования

#### 1.1.1 Система рекомендаций

Система рекомендаций представляет собой приложение, включающее поисковую строку и возможность выбора базы документов, по которой осуществлять поиск (рис. 1.1).

Рассматриваемая система имеет юридическую специфику, то есть предполагает работу с запросами и документами правовой направленности.

Сервис  ищет судебные акты по тексту или фрагменту искового заявления, претензии, решения госоргана, других документов или просто по описанию ситуации. Вы получите судебные решения, наиболее соответствующие обстоятельствам, изложенным в тексте.

Пожалуйста, учитывайте, что [параметры работы сервиса](#) могут быть изменены.

*Скопируйте и вставьте сюда текст либо опишите ситуацию своими словами. Чем подробнее вы изложите обстоятельства, тем точнее получится подборка.*

*Далее выберите, какие решения искать: арбитражных судов или судов общей юрисдикции.*

Найти акты арбитражных судов

Найти акты судов общей юрисдикции

Рис. 1.1 – Стартовая страница сервиса

По запросу пользователя система должна найти наиболее семантически близкие запросу документы и показать их в поисковой выдаче (рис. 1.2) в порядке убывания показателя семантической близости между тек-



стом запроса и текстом документа.

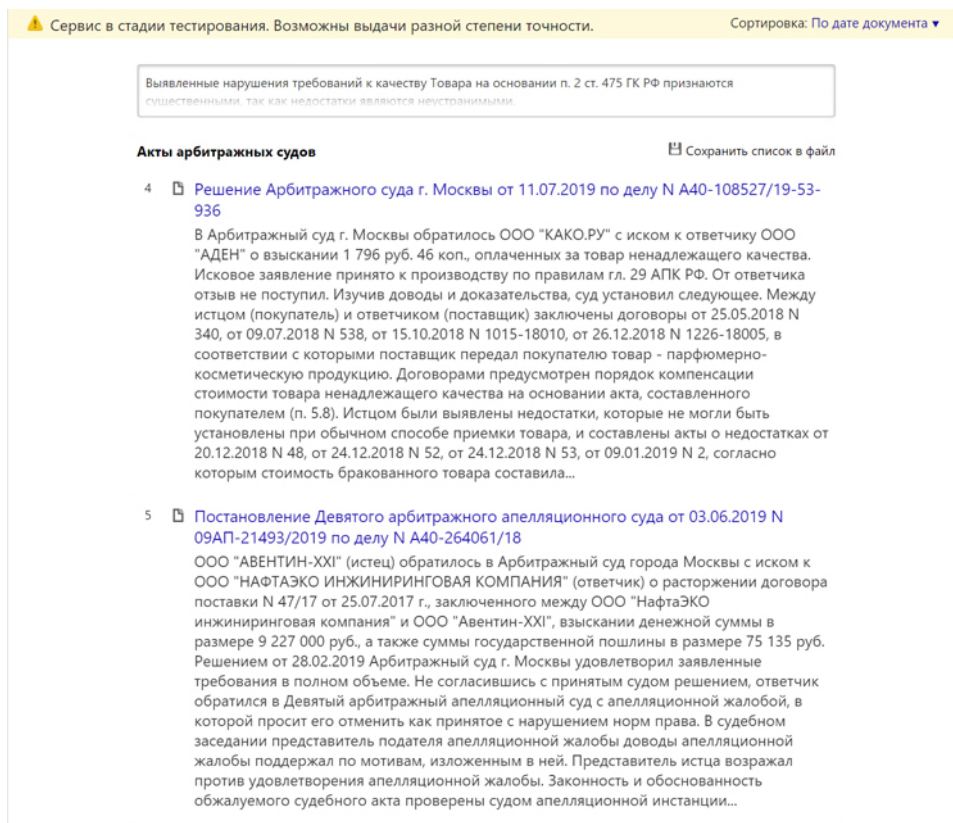


Рис. 1.2 – Поисковая выдача по запросу

Таким образом, ожидается, что по запросу пользователя будет получен список ранжированных документов, где первый документ является наиболее близким запросу по смыслу, а каждый последующий имеет меньшую семантическую близость с запросом, чем предыдущий.

### 1.1.2 Запрос

Запрос в системе – это описание какого-либо правового случая (рис. 1.3), для которого необходимо найти похожие правовые акты. Описание может осуществляться двумя основными способами:

1. Копирование текста или его части из реального юридического документа;
2. Описание ситуации своими словами.

Скопированный текст, как правило, отличается своим значительным

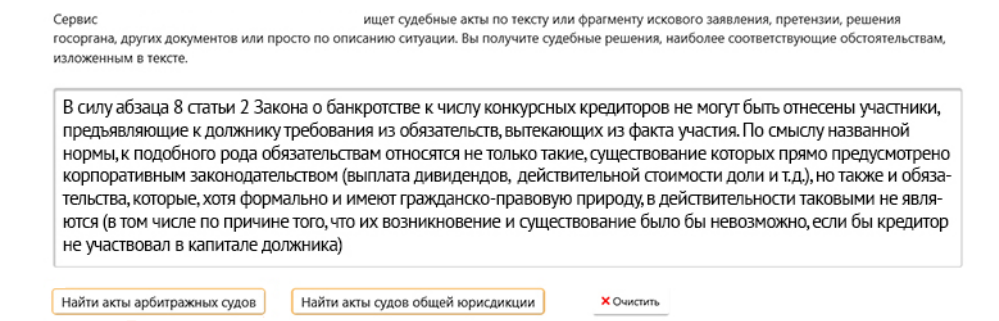


Рис. 1.3 – Пример запроса в системе

объемом, наличием специфических лексических оборотов, содержанием точной информации, такой как наименования объектов, ФИО участников ситуации, даты, денежные суммы, адреса и т.д.

При описании ситуации своими словами пользователи, как правило, стараются максимально кратко изложить суть, при этом используют более бытовые лексические конструкции, а также склонны заменять названия, ФИО и адреса синонимичными выражениями или цензорными оборотами.

Также запросы делятся на два вида по сфере описываемой ситуации:

1. Ситуации в сфере экономической деятельности;
2. Ситуации общей юрисдикции, где под общей юрисдикцией понимаются вопросы, связанные с гражданскими, административными и уголовными отношениями субъектов описываемой ситуации.

### 1.1.3 Рекомендация

Рекомендация в системе представляет собой реальный правовой акт, подходящий по смыслу ситуации, описанной в запросе. Рекомендации должны быть отсортированы в порядке убывания показателей их семантической близости с текстом запроса.

Пользователи могут прямо и косвенно оценивать предлагаемые им рекомендации. Оценить прямым способом пользователь может, ответив на

вопрос, подходит ли документ запросу (рис. 1.4).

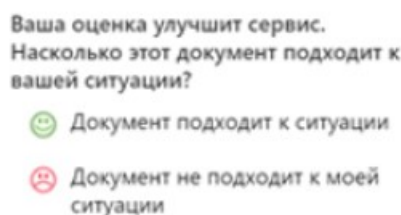


Рис. 1.4 – Инструмент для оценки рекомендации

Косвенной оценкой можно считать открытие документа, так как открываемый документ, как минимум, заинтересовал пользователя больше, чем неоткрытые документы.

## 1.2 Оценка работоспособности системы

Для оценки качества работы системы рекомендаций можно использовать поведение пользователей, а именно их оценки, описанные в 1.1.3.

Реализации обеих разновидностей оценок можно считать исходом опыта по схеме Бернулли [8]. При этом успехом в случае прямой оценки считается положительная оценка пользователя, а в случае косвенной - открытие документа из выдачи по запросу (табл. 1.1). Таким образом, можно видеть, что при прямой оценке происходит оценивание качества на уровне документов-рекомендаций, а при косвенной - на уровне запроса.

Таблица 1.1: Исход в опыте по оцениванию рекомендаций

	Реализация прямой оценки	Реализация косвенной оценки
Успех	Документ оценен положительно	По запросу открыт документ из выдачи
Неудача	Документ оценен отрицательно	По запросу нет открытий документов

Оба подхода к оценке успеха рекомендации можно использовать для

проверки гипотезы, что для различных групп запросов алгоритм рекомендаций работает одинаково. Группировку запросов можно проводить по способу задания запроса (копирование текста или описание своими словами), по сфере запроса, а также комбинируя группировки.

Отличительной чертой копируемых текстов является их длина, поэтому была выбрана граница длины запроса, после преодоления которой запрос считается скопированным текстом.

Таким образом, запросы группируются по длине:

1. Короткие (не более 70 слов);
2. Длинные (более 70 слов).

А также по сфере запроса:

1. Экономические;
2. В сфере общей юрисдикции.

Для каждого типа оценки (прямой и косвенной) была сформулирована гипотеза о равенстве долей [8] рекомендаций с успешным исходом для различных групп запросов (по длине запроса и сфере описанной в запросе ситуации).

В качестве нулевой гипотезы используется утверждение: вероятности успеха для обеих рассматриваемых групп запросов равны  $p$  (1.1). Альтернативная гипотеза является отрицанием нулевой (1.2).

$$H_0 : p_1 = p_2 = p \quad (1.1)$$

$$H_A : p_1 \neq p_2 \quad (1.2)$$

где  $p_i$  - вероятность успеха для  $i$  группы ( $i = 1, 2$ ).

Таким образом, сформулирована гипотеза (1.1)-(1.2): для различных групп запросов алгоритм рекомендаций работает одинаково.

### 1.3 Функционалы качества рекомендаций

Задача формирования рекомендаций – задача ранжирования. Для оценки качества ранжирования необходима специфическая разметка данных [1], где для каждой рекомендации в ранжированном списке ставится в соответствие некое значение, показывающее релевантность рекомендации запросу. На основе подобной разметки вычисляются такие показатели качества, как ...

В данной же работе рассматривается задача ранжирования в постановке, когда отсутствует такая разметка данных, поэтому стандартные способы оценки качества предлагаемых алгоритмов в таких условиях неприменимы.

В данной работе предлагается набор альтернативных функционалов качества ранжирования, не требующих разметку [1]. Новые функционалы качества используют информацию о прямых оценках пользователей документам, описанных в 1.1.3. Далее под разметкой будет пониматься набор данных запрос-документ, где каждый документ имеет оценку от пользователей. При этом важно, чтобы в таком наборе данных присутствовали только те запросы, для которых существует хотя бы одна релевантная и одна нерелевантная рекомендации.

На основе оценок пользователей предлагается оценивать показатели:

1. Правильности сортировки;
2. Ширины «окна» между релевантными и нет рекомендациями;
3. Степени поднятия релевантных документов.

Центральная идея новых функционалов - алгоритм рекомендаций с более высоким качеством ранжирования должен поднимать выше документы, которые пользователи считают релевантными, и опускать ниже нерелевантными.

levantные.

Показатель правильности сортировки представляет собой оцененную долю запросов, для которых алгоритм ставит все релевантные документы выше, а все нерелевантные - ниже. Данный показатель описывается формулой:

$$correctness = \frac{q_{good}}{|Q|} \quad (1.3)$$

где  $q_{good}$  - количество запросов, все релевантные документы которых выше нерелевантных,  $Q$  - множество запросов в разметке.

Под шириной окна между релевантными и нет рекомендациями понимается разница между максимальным номером позиции релевантного документа и минимальным номером позиции нерелевантного. Такое окно вычисляется для каждого запроса из разметки. Далее проводится статистический анализ полученных значений, в том числе оценивается некий статистический показатель. В данной работе предлагается оценивать медианный показатель:

$$window = median\left\{ \max_{good \in recom_q} pos(good) - \min_{bad \in recom_q} pos(bad), \quad q \in Q \right\} \quad (1.4)$$

где  $q$  - запрос,  $recom_q$  - множество рекомендаций по запросу  $q$ ,  $good$  - релевантная рекомендация,  $bad$  - нерелевантная рекомендация,  $pos(\cdot)$  - номер позиции рекомендации.

Степень поднятия релевантных документов можно считать дополнительной вспомогательной характеристикой алгоритма рекомендаций. Данный показатель - доля релевантных документов, которые подняты выше относительно своих позиций в рекомендациях старого алгоритма:

$$uplift = \frac{|doc : pos_{new}(doc) < pos_{old}(doc), \quad doc \in good_{doc}|}{|good_{doc}|} \quad (1.5)$$

где  $good_{doc}$  - множество релевантных рекомендаций,  $doc$  - рекомендация,  $pos_{new}(\cdot)$  - номер позиции в новых рекомендациях,  $pos_{old}(\cdot)$  - номер позиции

в старых рекомендациях.

В отличие от описанных выше функционалов, данный показатель в общем случае не показывает, насколько ранжирование нового алгоритма качественнее, чем ранжирование старого, однако он может быть полезен для понимания, работает ли новый алгоритм на том же множестве документов, что и старый.

Таким образом, предлагаемые функционалы качества имеют характеристику, представленную в таблице 1.2.

Таблица 1.2: Характеристика функционалов качества рекомендаций

Правильность сортировки	
Определение	Доля запросов, для которых все релевантные рекомендации находятся выше всех нерелевантных
Смысл	Показывает степень правильности сопоставления алгоритмом смысла запроса и смысла рекомендуемых документов
Недостатки	Отсутствует градация степени правильности сортировки, то есть показатель не учитывает случаи, когда практически все релевантные рекомендации оказались выше, но имеет место незначительное количество нерелевантных рекомендаций, оказавшихся выше
Ширина окна	
Определение	Медиана разницы наибольшей позиции релевантного документа и наименьшей позиции нерелевантного

Продолжение таблицы 1.2

Смысл	Показывает, насколько далеко позиционно находятся группа релевантных и группа нерелевантных документов
Недостатки	Показатель не учитывает случаи, когда практически все релевантные рекомендации оказались выше, но имеет место незначительное количество нерелевантных рекомендаций, оказавшихся выше
Степень поднятия	
Определение	Доля релевантных документов, которые были подняты выше относительно позиционирования базовым алгоритмом
Смысл	Показывает, какое количество релевантных документов пользователи увидят раньше, чем при базовом ранжировании
Недостатки	Не учитывает, что новые алгоритмы могут не поднимать релевантные документы по причине нахождения более релевантных

#### 1.4 Рекомендательные модели

В качестве потенциальных рекомендательных алгоритмов были рассмотрены как предобученные [4, 7] модели, так и спроектированные лингвистические и семантические модели.

Лингвистические модели более просты в реализации, однако не используют информацию о семантике текста, а опираются только на различные частоты встречаемости слов.



Семантические же модели более сложные в проектировании, так как требуют обучения, однако они учатся понимать смысл текстов. Также можно заранее создавать базы данных для семантических векторов документов, что значительно сокращает затраты по времени на обработку пар запрос-документ, так как требует лишь разовой обработки запроса.

Рассматриваемые семантические модели делятся на следующие группы:

1. Классификаторы соответствия документа запросу
2. Рекомендации по векторной близости документа запросу

#### 1.4.1 Лингвистические алгоритмы

Основная идея лингвистических алгоритмов - идея построения рекомендаций на основе ранжирования документов по различным показателям пересечения слов запроса и документа.

Были рассмотрены следующие лингвистические алгоритмы рекомендаций:

1. Рекомендации по пересечению слов запроса и документа
2. Рекомендации по сумме IDF [2] слов запроса
3. Рекомендации по сумме TF-IDF [2] слов запроса
4. Рекомендации по сумме BM25 [3] слов запроса

Все эти алгоритмы имеют одну и ту же схему построения рекомендаций, которую можно описать рисунком 1.5.

Пусть  $Q_{word}$  - множество слова запроса;  $stop$  - множество стоп-слов;  $D_{word}$  - множество слов документа;  $N$  - количество документов;  $df_{word}$  - количество документов, содержащих  $word$ ;  $k, b$  - выбираемые параметры;  $avgdl$  - среднее количество слов в документе.

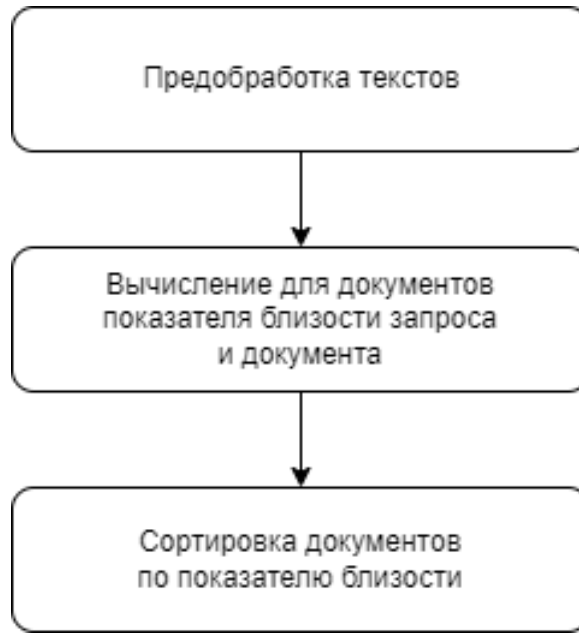


Рис. 1.5 – Лингвистический алгоритм рекомендаций

Также вводится понятие индикаторной функции, которая показывает наличие или отсутствие слова во множестве слов:

$$I_{\in}(w, d) = \begin{cases} 1, & w \in d \\ 0, & w \notin d \end{cases}$$

а также индикаторной функции, сигнализирующей о совпадении слов:

$$I(w, word) = \begin{cases} 1, & w = word \\ 0, & w \neq word \end{cases}$$

В [2] описаны показатели TF-IDF, которые ориентируются на показатели частотности слов в запросе и документах:

$$TF(word, doc) = \frac{\sum_{d \in doc} I(d, word)}{|doc|}$$

$$IDF_1(word) = \log \frac{N}{df_{word}} \quad (1.6)$$

В [3] предлагается альтернативное представление IDF:

$$IDF_2(word) = \log \frac{N - df_{word} + 0.5}{df_{word} + 0.5} \quad (1.7)$$

Показатель близости при построении рекомендаций по пересечению слов выглядит следующим образом:

$$similarity_{\cap} = \frac{\sum_{word \in Q_{word} \setminus stop} I_{\in}(word, D_{word})}{|Q_{word} \setminus stop|} \quad (1.8)$$

При сортировке по сумме IDF показателем близости является:

$$similarity_{IDF} = \sum_{word \in (D_{word} \cap Q_{word}) \setminus stop} IDF_1(word) \quad (1.9)$$

Показателем близости при сортировке по сумме TF-IDF является:

$$similarity_{TF-IDF} = \sum_{word \in (D_{word} \cap Q_{word}) \setminus stop} TF(word, D_{word}) IDF_1(word) \quad (1.10)$$

Показатель близости при построении рекомендаций по сумме BM25 слов:

$$similarity_{BM25} = \sum_{word \in (D_{word} \cap Q_{word}) \setminus stop} \frac{TF(word, D_{word})(k+1)}{TF(word, D_{word}) + k(1-b + \frac{|D_{word}|}{avgdl})} \cdot IDF_2(word) \quad (1.11)$$

Таким образом, схема 1.5 и показатели близости (1.8)-(1.11) представляют собой соответствующие лингвистические алгоритмы.

Главным недостатком лингвистических алгоритмов является наличие вычислительных и временных затрат для каждой пары запрос-документ, которые могут быть достаточно большими, так как невозможно заранее создать базу данных лингвистических признаков документов, ведь они становятся известными непосредственно в связке с текстом запроса.

#### 1.4.2 Классификаторы соответствия документа запросу

Основная идея данного подхода - построить классификатор для пар векторных представлений текстов запроса и документа. В задаче присутствует два класса пар запрос-документы: документ подходит запросу и документ не подходит запросу. В качестве классификаторов рассмотрены:

1. kNN-классификатор семантических векторов [5, 7]
2. LightGBM-классификатор показателей векторной близости [6, 7]
3. BERT-классификатор [4, 7]

В основе первых двух классификаторов лежит идея использования результатов работы предобученной модели с архитектурой BERT [4]. Получаемые векторные представления используются в качестве входных данных моделей машинного обучения kNN и LightGBM. Таким образом, общая схема работы моделей выглядит следующим образом:

1. Получение векторных представлений текстов запроса и документа с помощью предобученной модели;
2. Обработка полученных представлений, формирование признаков классификатора;
3. Классификация пары запрос-документ по степени соответствия документа запросу.

Для kNN-классификатора обработка векторных представлений сводится к их конкатенации. При такой обработке представлений выбор модели обусловлен тем, что он направлен на поиск ближайших соседей в метрическом пространстве, а семантические векторы образуют метрическое пространство.

Для LightGBM-классификатора обработка семантических векторов представляет собой формирование набора значений показателей близости векторов текстов запроса и документа. Соответствующие показатели близости приведены в таблице 1.3. Так как показатели близости часто могут иметь различные области значений, то в качестве модели используется градиентный бустинг над деревьями решений.

BERT-классификатор представляет собой предобученный BERT с до-

обучаемой на задачу классификации головой. Отличие данной модели от двух предыдущих в том, что она не является дополнительной надстройкой над результатом работы предобученной модели, а ее признаками являются сами тексты запросов и документов, обработанные особым образом.

Таким образом, модели классификаторы имеют разную архитектуру и разные наборы признаков, представленные в таблице 1.3.

Таблица 1.3: Классификаторы соответствия документа запросу

Архитектура модели	Входные признаки модели
Предобученный BERT + kNN	Сконкатенированные векторные представления текстов запроса и документа, полученные предобученной моделью
Предобученный BERT + LightGBM классификатор	Показатели близости векторного представления документа и запроса: несходство Брея — Кертиса; расстояние Канберра; расстояние Чебышева; Манхэттенское расстояние; коэффициент корреляции; косинусовое расстояние; евклидово расстояние; расстояние Минковского
Предобученный BERT + голова на классификацию	Текст запроса и текст документа со специальными токенами

Итоговая сортировка документов осуществляется по показателю вероятности принадлежности документа классу подходящих запросу. Соответственно, в итоговом ранжированном списке документы отсортированы по убыванию данного показателя. Показатель вероятности является выходом

модели и представляет собой значение сигмоидальной функции (1.12).

$$\sigma(x) = \frac{1}{1 + \exp^{-x}} \quad (1.12)$$

### 1.4.3 Рекомендации по векторной близости документа запросу

Идея данного подхода - использование дообученной модели для получения векторных представлений текста запроса и текста документа и дальнейшее ранжирование документов по показателю близости между полученными векторами.

Для дообучения берется предобученная модель с архитектурой BERT (Bidirectional Encoder Representation Transformers) [4] - модель представления языка, которая предназначена для предварительного обучения глубоких двунаправленных представлений на простых немаркированных текстах путем совмещения левого и правого контекстов во всех слоях. Это позволяет настраивать предварительно обученную модель BERT с помощью лишь одного дополнительного выходного слоя и получать наиболее актуальные результаты для широкого спектра задач.

Проще говоря, дообучение осуществляется путем решения с помощью модели двух задач классификации [4]:

1. NSP (Next Sentence Prediction) - классификация пары предложений на предмет их последовательного расположения в тексте;
2. MLM (Masked-Language Modeling) - предсказание скрытого токена по контексту.

При обучении модель пытается как можно лучше научиться решить данные задачи, что приводит к получению достаточно качественных семантических векторов для подающихся на вход текстов.

Так как данная модель учится «понимать» смысл специфических юридических текстов, то в качестве показателя семантической близости меж-

ду запросом и документом можно использовать косинусову меру близости между получаемыми векторами запроса и документа:

$$similarity_{cos} = 1 - \cos(query, doc) \quad (1.13)$$

где *query* - вектор текста запроса, *doc* - вектор текста документа.

Ранжированные списки документов по запросу получаются с помощью (1.13) путем сортировки показателя по убыванию его значения.

### 1.5 Предобработка текстовых данных

Для лингвистических моделей предобработка текста сводится к:

1. Удалению знаков препинания;
2. Разделению текста на слова и их приведение к некой начальной форме;
3. Формированию словаря при исключении из него стоп-слов (наиболее частотных слов, вспомогательных слов, союзов и предлогов).

Для kNN и LightGBM классификаторов предварительная обработка не требуется, так как все необходимые для обучения и работы моделей признаки выделяются с помощью сторонней предобученной модели.

Входными данными для BERT-классификатора является последовательность токенов, которые сначала преобразуются в векторы, а затем обрабатываются в нейронной сети. Но перед тем, как обработка может начаться, необходимо обработать входные данные и украсить их некоторыми дополнительными метаданными:

1. Встраивание токенов: токен [CLS] добавляется к токенам входного слова в начале первого предложения, а токен [SEP] вставляется в конце каждого предложения;
2. Встраивание сегмента: к каждому токenu добавляется маркер, обо-

значающий предложение А или предложение В. Это позволяет кодировщику различать предложения;

3. Позиционные вложения. Позиционные вложения добавляются к каждому токenu, чтобы указать его положение в предложении.

Токенизация выполняется с помощью соответствующего предобученного токенизатора. В качестве предложения А предлагается использовать текст запроса, а в качестве предложения В - текст документа.

По итогу такой обработки получаются следующие векторы, которые в дальнейшем подаются на вход модели:

1. Вектор индексов токенов вида «[CLS] текст запроса [SEP] текст документа [SEP]», который при необходимости в конце дополняется специальными токенами [PAD], обозначающими дозаполнение вектора токена до максимальной длины;

2. Вектор внимания, в котором на позициях токенов [PAD] находятся 0, а на позициях остальных токенов - 1;

3. Вектор типа токенов, в котором на позициях токенов предложения А находятся 1, а на позициях предложения В - 0.

При дообучении языковой модели к описанным выше векторам добавляются еще два: копия вектора индексов токенов и вектор метки характера последовательности предложений, а также оригинальный вектор индексов токенов преобразовывается под задачу MLM путем замены случайных токенов специальным токеном [MASK]. Однако в качестве исходных текстов для обработки необходимо использовать корпус текстов документов, для которого в качестве предложений А и В берутся реальные предложения из текстов.

Для получения разметки пар предложений на предмет их последова-



тельности используется алгоритм, представленный на рис. 1.6.

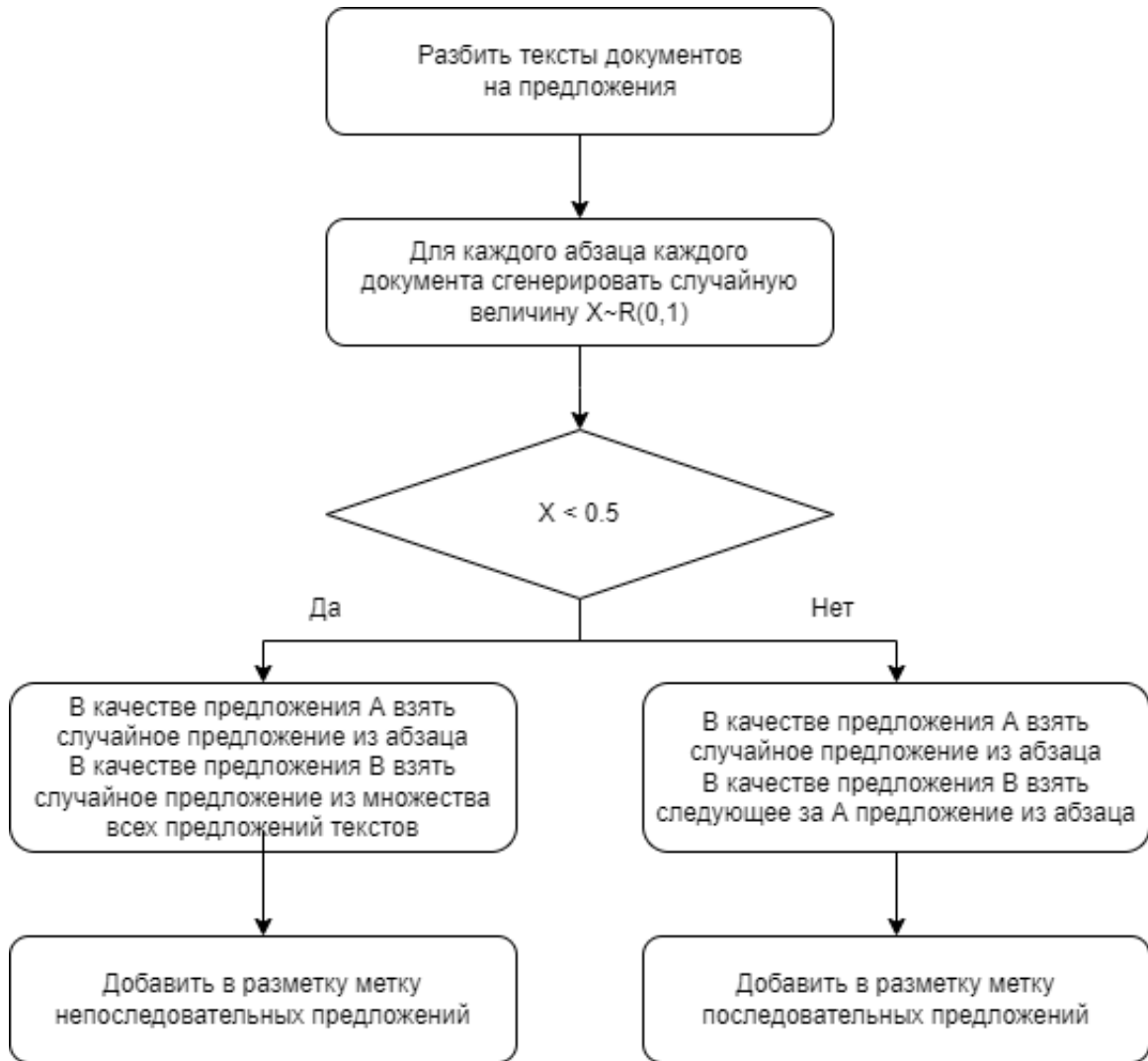


Рис. 1.6 – Алгоритм генерации NSP-разметки

## 1.6 Постановка задачи оценки вероятности обслуживания преследуемого игрока

Будем рассматривать случайную величину  $\xi$ , представляющую собой индикаторную функцию события  $A = \{\text{В результате игровой ситуации произошло обслуживание преследуемого объекта преследователем}\}$ :

$$\xi = \begin{cases} 1, \text{событие } A \text{ произошло} \\ 0, \text{событие } A \text{ не произошло} \end{cases} \quad (1.14)$$

и случайную величину терминального промаха  $\eta = h(\vec{\Theta})$ , где  $\vec{\Theta}$  - вектор случайных координат преследователя с распределением (??), а  $h(\vec{\Theta})$  - некая

функция, значение которой вычисляется в процессе моделирования процесса игры.

Для рассматриваемых случайных величин можно определить условное математическое ожидание I и II типов [9].

Условное математическое ожидание I типа представляет собой некоторую функцию, принимающую детерминированные значения:

$$M[\xi|\eta = d] = R(d) \quad (1.15)$$

В (1.15) описано математическое ожидание случайной величины  $\xi$  при условии, что случайная величина  $\eta$ , измеренная в опыте, приняла значение  $d$ .

Условное математическое ожидание II типа - случайная величина  $V$ :

$$V = M[\xi|\eta] = R(\eta) \quad (1.16)$$

Для (1.16) справедлива формула полного математического ожидания [9]:

$$M[\xi] = M[M[\xi|\eta]] \quad (1.17)$$

В то же время, нетрудно показать, что:

$$M[\xi] = P(A), \quad (1.18)$$

так как  $\xi$  является индикаторной функцией для события  $A$ , следовательно, среднее количество произошедших событий  $A$  в серии опытов - вероятность события  $A$ .

Из (1.17) и (1.18) следует, что:

$$P(A) = M[M[\xi|\eta]], \quad (1.19)$$

то есть безусловная вероятность того, что в результате игровой ситуации произошло обслуживание преследуемого объекта преследователем, равна

значению полного математического ожидания случайной величины  $\xi$  относительно случайной величины  $\eta$ .

Оценку безусловной вероятности обслуживания можно получить с помощью метода Монте-Карло [?]. Для этого необходимо многократно разыграть игровую ситуацию между преследуемым и преследователем и получить реализации терминального промаха.

Пусть  $\vec{\theta}_k$  - реализация вектора случайных координат преследователя  $\vec{\Theta}$ , полученная в результате компьютерного моделирования. Тогда  $d_k = h(\vec{\theta}_k)$  - реализация терминального промаха, полученная в результате моделирования процесса игры при  $\vec{\theta}_k$ . Будем считать, что функция условного математического ожидания  $R(d)$  I типа задана. В этом случае,  $R_k = R(d_k)$  - реализации условной вероятности обслуживания. Тогда, согласно (1.17) и (1.19), оценка безусловной вероятности обслуживания имеет вид:

$$\hat{P}(A) = \hat{M}[R(d)] = \frac{R_1 + \dots + R_n}{n}, \quad (1.20)$$

где  $n$  - количество реализаций терминального промаха.

Таким образом, необходимо с помощью компьютерного моделирования оценить безусловную вероятность обслуживания преследуемого объекта в задаче (1.20).

### 1.7 Описание параметров задачи

Опираясь на все вышесказанное, исходными данными к задаче являются параметры для задачи проверки гипотез (1.1)-(1.2), задачи оценки доверительных интервалов оценок на разметке (2.1), а также параметры рекомендательных моделей.

Параметрами для проверки гипотез и задачи оценки доверительных интервалов оценок на разметке выступают уровни доверия.

Для лингвистических моделей единственными параметрами являются

ся коэффициенты  $k, b$ , используемые для вычисления показателя близости текстов запроса и документа (1.11) в алгоритме ранжирования по сумме BM25.

Параметрами для kNN выступают количество соседей  $k$  и степень функции расстояния Минковского  $p$ .

Параметрами для алгоритма LightGBM являются скорость обучения  $lr$ , количество листьев дерева  $leaves$ , максимальная глубина дерева  $depth$ , количество деревьев  $n$  в ансамбле, над которым проводится градиентный бустинг.

Параметрами для моделей BERT выступают количество эпох обучения  $epoch$ , размер батча  $batch$ , скорость обучения  $lr$  и коэффициент регуляризации  $L$ . Параметры количества эпох и размера батча выбираются исходя из объема обучающих данных и сложности модели. Скорость обучения и коэффициент регуляризации являются параметрами функции оптимизации.

## 2 ПРАКТИЧЕСКАЯ ЧАСТЬ

### 2.1 Алгоритм решения задачи

Для решения задачи сравнения эффективности алгоритмов рекомендаций семантически схожих документов необходимо предварительно решить несколько подзадач, а именно:

1. Проверить гипотезу (1.1)-(1.2) о равенстве долей успешных исходов в различных группах запросов. По полученным результатам сделать выводы о работоспособности системы;
2. Сформировать набор данных для оценки качества ранжирования и оценка погрешностей оценок на полученной разметке;
3. Обработать текстовые данные;
4. Реализовать рекомендательные модели;
5. Получить результаты ранжирования размеченных данных реализованными моделями;
6. По полученным результатам провести оценку качества рекомендаций с помощью (1.3)-(1.5).

При решении задачи используется язык программирования Python.

Для проверки сформулируем 8 гипотез о равенстве долей (по 4 гипотезы для каждого вида оценки: прямой и косвенной). В качестве групп для каждой гипотезы рассматриваем следующие пары множеств запросов: короткие и длинные, экономические и в сфере общей юрисдикции, короткие и длинные в экономической сфере, короткие и длинные в сфере общей юрисдикции. Для проверки гипотез используем библиотеку SciPy.

Для дальнейшего применения фнукционалов качества (1.3)-(1.5) фор-

мируем разметку данных, в которой для всех запросов которой существуют как положительно, так и отрицательно оцененные документы. Так как пользователи достаточно редко ставят и положительные, и отрицательные оценки по одному запросу, расширим понятие нерелевантной рекомендации. Таковой будет считаться документ с отрицательной оценкой или пропущенный документ.

Пропущенный документ – документ на 1-4 позициях, который не был открыт пользователем при том, что на позиции ниже были открытия документов. Ограничение по позициям обусловлено тем, что на экране поисковой выдачи помещаются 4 документа, то есть можно считать, что пользователь всегда видит первые 4 рекомендации.

Для полученной разметке с использованием библиотеки SciPy оцениваем доверительные интервалы [9] потенциальных оценок по формуле:

$$\delta = u_{1-\frac{\alpha}{2}} \sqrt{\frac{p(1-p)}{n} \left(1 - \frac{n}{N}\right)} \quad (2.1)$$

где  $p = 0.5$  - доля респондентов с оценкой,  $\alpha$  - уровень значимости,  $u_{1-\frac{\alpha}{2}}$  - квантиль уровня  $1 - \frac{\alpha}{2}$  стандартного распределения,  $n$  - количество респондентов,  $N$  - объем генеральной совокупности.

Формирование словаря для лингвистических моделей, а также предварительный расчет показателей IDF (??), (1.7) осуществляем на корпусе текстов документов. Обработку текстов производим с помощью библиотеки `rumorphy2`, которая позволяет выделять начальные формы слов средствами морфологического анализа слов. Множество стоп-слов определяем по графикам частотности - исключаем наиболее частотные слова.

Для моделей kNN и LightGBM предварительная обработка текстов не требуется.

При обработке текстов для BERT-классификатора предварительно вы-

бираем значение максимальной длины входных векторов и ограничиваем длину текста запроса и текста документа значением, равным половине от максимальной длины. Для каждой пары запрос-документ проводим токенизацию каждого из текстов и отсекаем лишние токены, которые не помещаются в ограничения по длине векторов. Полученные пары текстов токенизируем с помощью токенизатора соответствующей модели из библиотеки `transformers`.

Тексты для языковой модели обрабатываем по алгоритму рис. 1.6. При разделении текстов на предложения учитываем специфические особенности юридических текстов, а именно, что точка может использоваться не только для обозначения конца предложения, но и в качестве сокращения, например, в ФИО. По этой причине используем более сложный алгоритм разбиения на предложения (приложение ...

Проводим токенизацию с помощью библиотеки для каждой полученной пары предложений и далее маскируем 15% токенов специальным токеном [MASK].

Для лингвистических моделей реализуем функции вычисления показателей лингвистической близости (1.8)-(1.11) запроса и документа. Реализация соответствующих моделей рекомендаций далее заключается лишь в сортировке документов по вычисляемым показателям близости.

Для семантических моделей в качестве предобученной языковой модели возьмем LaBSE [10] с архитектурой BERT [4] из библиотеки `transformers`. Реализации алгоритмов kNN и LightGBM берем из библиотеки `sklearn`.

При построении классификаторов в качестве разметки данных будем использовать прямые оценки пользователей. На этапе обучения оценку качества соответствующих моделей будем проводить с помощью привычных функционалов качества для задачи классификации: показателей точности,

полноты и F1 для каждого класса, а также с помощью AUC под ROC-кривой. ¶ В случае kNN и LightGBM обучим только эти модели машинного обучения. В случае BERT-классификатора дообучим голову на задачу классификации.

Для получения рекомендаций по векторной близости документа запросу дообучим на корпусе текстов документов предобученную языковую модель, которая будет учиться прогнозировать пропущенное слово по контексту, а также понимать, является ли следующее предложение продолжением предыдущего. Для этого используем обработанный ранее корпус текстов документов.

После проектирования моделей и их обучения получим результаты ранжирования на сформированной разметке данных. Для лингвистических моделей ранжирование проводим по соответствующим показателям близости текстов (1.8)-(1.11). Для моделей классификации в качестве показателя близости используем значение сигмоидальной функции (1.12) класса релевантного документа, которая является выходом модели. Ранжирование при использовании языковой модели осуществляем по показателю косинусовой близости (1.13) между получаемыми от языковой модели векторными представлениями тестов запроса и документа. При построении рекомендаций дополняем основную разметку неразмеченными документами, которые будут использоваться только для разбора примеров рекомендаций и расширенного описания характеристик алгоритмов.

Для полученных рекомендаций используем функционалы качества (1.3)-(1.5) для их оценки и сравнения между собой.

## 2.2 Параметры задачи

Гипотезы (1.1)-(1.2) проверяются на уровне доверия 0.99. Оценка доверительных интервалов оценок на разметке (2.1) проводится на уровне



доверия 0.95.

Для вычисления показателя близости текстов запроса и документа (1.11) в алгоритме ранжирования по сумме BM25 используются параметры  $k = 0.3, b = 0$ .

Параметры моделей рекомендаций представлены в соответствующих таблицах.

Таблица 2.1: Параметры kNN-классификатора

Параметр	k	p
Значение	5	2

Таблица 2.2: Параметры LightGBM-классификатора

Параметр	lr	leaves	depth	n
Значение	0.1	31	-1	100

Таблица 2.3: Параметры BERT-классификатора

Параметр	epoch	batch	lr	L
Значение	4	32	0.00002	0

Таблица 2.4: Параметры языковой модели

Параметр	epoch	batch	lr	L
Значение	1	1	0.00002	0

В качестве функции оптимизации для BERT-моделей используется AdamW, при этом скорость обучения линейно уменьшается после каждого шага обучения.

### 2.3 Результаты численного эксперимента

Результаты проверки гипотез (1.1)-(1.2) приведены в приложении 1. Можно видеть, что для всех групп запросов гипотеза была отвергнута. При этом для ряда случаев статистика критерия значительно больше, чем квантиль соответствующего уровня.

Таким образом, по совокупности проверенных гипотез, алгоритм работает не одинаково для различных групп запросов, то есть настоящий алгоритм может быть улучшен и рассматриваемая в данной работе задача имеет смысл.

Результаты оценки доверительных интервалов на сформированной разметке представлены в приложении 2. Для всех групп запросов и релевантных документов погрешность оценок на сформированной разметке не должна превышать 5 процентных пункта, то есть оценки будут достаточно точными.

На этапе обучения оценка качества классификаторов производилась с помощью привычных функционалов качества для задачи классификации. В приложениях 4-9 представлены результаты классификации на отложенной выборке для соответствующих моделей. Первый класс - класс документов, которые не подходят запросу. Второй класс - класс подходящих запросу документов.

Достаточно похожими по качеству классификации являются модели kNN (приложения 4-5) и LightGBM (приложения 6-7), хотя вторая модель по показателям и дает незначительный прирост качества.

Наиболее точная классификация получается с помощью классификатора BERT (приложения 8-9), однако при сравнении точности на обучающей и валидационной выборке можно увидеть, что эта модель склонна к переобучению.

Таким образом, обученные классификаторы показывают достаточно неплохое качество на задаче классификации.

На рисунках 2.1, 2.2 представлена оценка качества ранжирования документов различными алгоритмами. Оценки были получены для различных групп запросов с помощью функционалов (1.3)-(1.5).

	Текущий алгоритм	$\cap$ слов	$\sum IDF$	$\sum TFIDF$	$\sum BM25$
ОЦЕНКА ПРАВИЛЬНОСТИ СОРТИРОВКИ					
Все запросы	0.15	0.29	0.3	0.3	0.3
Экономические запросы	0.17	0.28	0.29	0.27	0.28
Запросы общей юрисдикции	0.12	0.3	0.32	0.34	0.33
Короткие запросы	0.14	0.28	0.29	0.31	0.28
Длинные запросы	0.16	0.27	0.32	0.29	0.32
МЕДИАНА РАЗНИЦЫ ПОЗИЦИЙ РЕЛЕВАНТНЫХ И НЕТ РЕКОМЕНДАЦИЙ					
Все запросы	-7	-2	-3	-5	-3
Экономические запросы	-5	-9	-15	-13	-15
Запросы общей юрисдикции	-10	-1	-1	-1	-1
Короткие запросы	-7	-3	-2	-3	-3
Длинные запросы	-7	-2	-5	-8	-5
ОЦЕНКА ПОДНЯТИЯ РЕЛЕВАНТНЫХ ДОКУМЕНТОВ					
% поднятых	-	25%	30%	28%	30%

Рис. 2.1 – Оценка качества лингвистических алгоритмов

По полученным результатам можно сделать следующие выводы по задаче:

1. Лингвистические алгоритмы поднимают преимущественно те документы, которые были найдены текущим алгоритмом;
2. Семантические алгоритмы поднимают преимущественно незамеченные текущим алгоритмом документы, которые не были оценены пользователями ввиду своей недоступности;
3. Наилучшие результаты показывают BERT и LightGBM классификаторы;
4. Все новые алгоритмы по показателю правильности сортировки луч-

	Текущий алгоритм	kNN	LightGBM	BERT- классификатор	Языковая модель BERT
ОЦЕНКА ПРАВИЛЬНОСТИ СОРТИРОВКИ					
Все запросы	0.15	0.22	0.45	0.68	0.3
Экономические запросы	0.17	0.21	0.46	0.66	0.26
Запросы общей юрисдикции	0.12	0.24	0.45	0.72	0.31
Короткие запросы	0.14	0.22	0.43	0.75	0.29
Длинные запросы	0.16	0.23	0.48	0.67	0.26
МЕДИАНА РАЗНИЦЫ ПОЗИЦИЙ РЕЛЕВАНТНЫХ И НЕТ РЕКОМЕНДАЦИЙ					
Все запросы	-7	0	0	6	0
Экономические запросы	-5	0	0	6	0
Запросы общей юрисдикции	-10	0	0	6	0
Короткие запросы	-7	0	0	5	0
Длинные запросы	-7	0	0	6	0
ОЦЕНКА ПОДНЯТИЯ РЕЛЕВАНТНЫХ ДОКУМЕНТОВ					
% поднятых	-	6%	9%	45%	11%

Рис. 2.2 – Оценка качества семантических алгоритмов

ше, чем текущий алгоритм;

5. Предложенные функционалы качества позволяют сравнивать между собой ранжирование алгоритмов, однако не является исчерпывающей характеристикой алгоритмов.

## ЗАКЛЮЧЕНИЕ

В данной выпускной квалификационной работе предлагается исследование подходов к решению задачи оценки безусловной вероятности обслуживания в терминах описанного игрового процесса.

Были выбраны и математически описаны модели движения участников игры: преследуемый объект совершает движение в соответствии с уравнениями движения в области с действующими силами тяжести и сопротивления воздуха, преследователь движется в соответствии с идеальным методом наведения на фактическое место встречи с маневрирующей целью. В основу моделирования случайного местоположения на плоскости легли идеи метода Неймана, а также был разработан собственный алгоритм распределения случайных координат для сети дорог. Было произведено компьютерное моделирование игровой ситуации, и на основе результатов компьютерного моделирования с применением метода Монте-Карло были получены оценки безусловной вероятности обслуживания для данной постановки задачи.

Численные результаты эксперимента показали, что оценка вероятности, полученная с помощью метода Монте-Карло, сходится при стремлении количества реализаций к бесконечности. Также была подтверждена зависимость оценки вероятности обслуживания от параметров движения преследуемого.

Разработанный алгоритм имеет линейную зависимость времени работы от количества требуемых реализаций, что является одним из наиболее эффективных показателей оптимальности программы.

В работе приведены обзоры на различные способы решения проблемы для более сложных постановок задачи, и полученные результаты в пер-

спективе могут быть обобщены на более общие случаи.

## ЛИТЕРАТУРА

- [1] Tie-Yan Liu Learning to Rank for Information Retrieval. — Foundations and Trends in Information Retrieval: Vol. 3: No 3, с. 225—331. — 2009. — 103 с.
- [2] Salton, G. and Buckley, C. Term-weighting approaches in automatic text retrieval. — Information Processing Management. 24(5): 513—523 с. — 1988.
- [3] Nick Craswell, Hugo Zaragoza, Stephen Robertson. In Proceedings of the Fourteenth Text REtrieval Conference. — Microsoft Cambridge at TREC-14: Enterprise Track. (TREC 2005). — Gaithersburg, USA — 2005.
- [4] Polosukhin, Illia; Kaiser, Lukasz; Gomez, Aidan N.; Jones, Llion; Uszkoreit, Jakob; Parmar, Niki; Shazeer, Noam; Vaswani, Ashish. Attention Is All You Need. — 2017. — 15 с.
- [5] Daniel T. Larose. Discovering Knowledge in Data: An Introduction to Data Mining. — 2004. — 90-106 с.
- [6] Sagi, Omer; Rokach, Lior. Approximating XGBoost with an interpretable decision tree. — Information Sciences. 572 — 2021. — 522-542 с.
- [7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. — 2019. — 16 с.
- [8] Ивченко Г. И., Медведев Ю. И. Введение в математическую статистику. — М.: Издательство ЛКИ — 2010. — 600 с.
- [9] А. И. Кибзун, Е. Р. Горяинова, А. В. Наумов, А. Н. Сиротин ТЕОРИЯ ВЕРОЯТНОСТЕЙ И МАТЕМАТИЧЕСКАЯ СТАТИСТИКА. БАЗО-

ВЫЙ КУРС С ПРИМЕРАМИ И ЗАДАЧАМИ – М.: ФИЗМАТЛИТ,  
2002. — 224 с.

- [10] Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, Wei Wang  
Language-agnostic BERT Sentence Embedding. — Google AI Mountain  
View — 2020. — 14 с.



## ПРИЛОЖЕНИЯ

## Приложение 1.

## Оценка работоспособности системы

Группировка	Статистика критерия	p-value	Гипотеза	Вердикт
Прямая оценка				
По сфере запроса	7.25	0	Отвергнута	Для запросов экономического характера положительная оценка более вероятна
По длине запроса	23.41	0	Отвергнута	Для длинных запросов положительная оценка значительно более вероятна
По длине запроса в экономической сфере	10.09	0	Отвергнута	Для длинных запросов положительная оценка более вероятна
По длине запроса в сфере общей юрисдикции	20.33	0	Отвергнута	Для длинных запросов положительная оценка значительно более вероятна
Косвенная оценка				

По сфере запроса	95.39	0	Отвергнута	Для запросов экономического характера открытие рекомендации существенно более вероятно
По длине запроса	40.19	0	Отвергнута	Для длинных запросов открытие рекомендации существенно более вероятно
По длине запроса в экономической сфере	8.82	0	Отвергнута	Для длинных запросов открытие рекомендации более вероятно
По длине запроса в сфере общей юрисдикции	44.53	0	Отвергнута	Для длинных запросов открытие рекомендации существенно более вероятно

## Приложение 2.

Оценка погрешности при оценивании доли запросов с правильной  
сортировкой

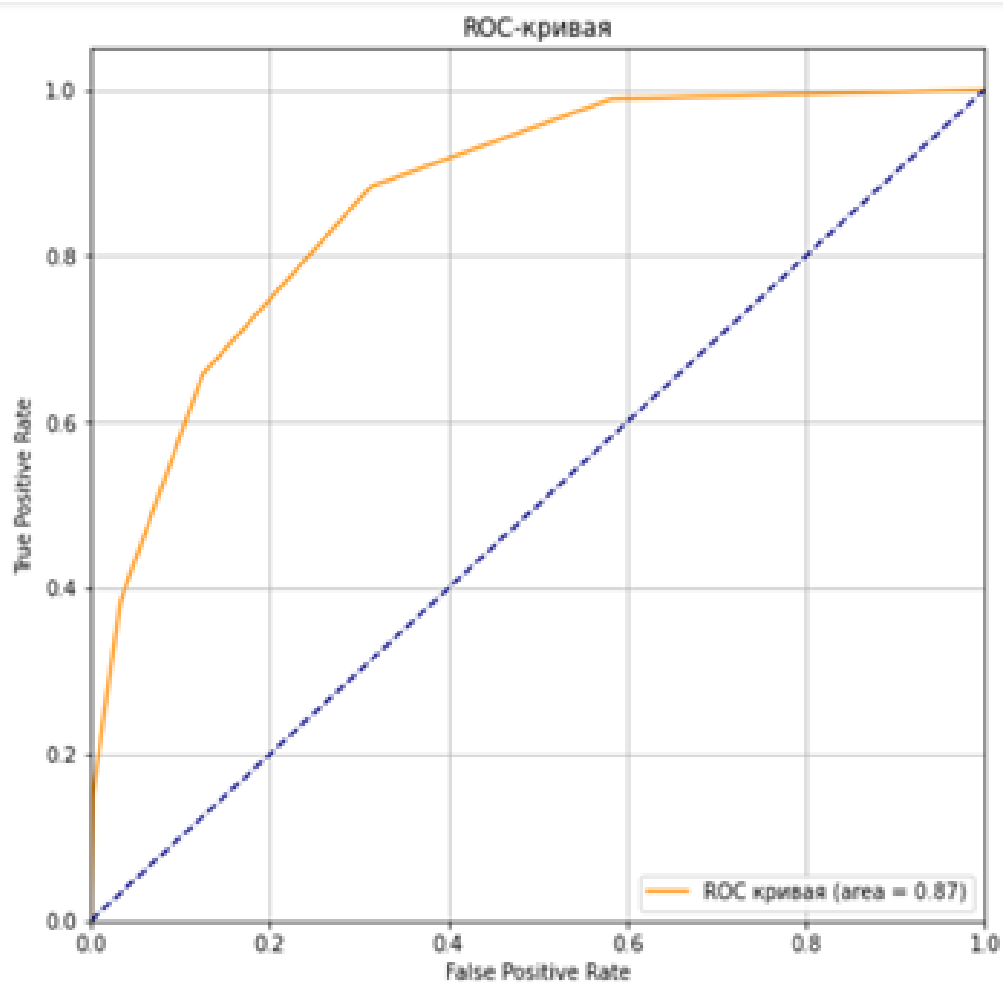
Запросы	Объем раз- метки	Погрешность (п.п.)
Все	744	+ -2.93
В сфере экономической дея- тельности	447	+ -3.59
В сфере общей юрисдикции	297	+ -4.41
Длинные запросы	398	+ -3.35
Короткие запросы	346	+ -3.86

## Приложение 3.

Оценка погрешности при оценивании доли поднятых релевантных  
документов

Документы	Объем раз- метки	Погрешность (п.п.)
Релевантные документы	2726	+ -1.56

## ROC-кривая качества классификации kNN-классификатора

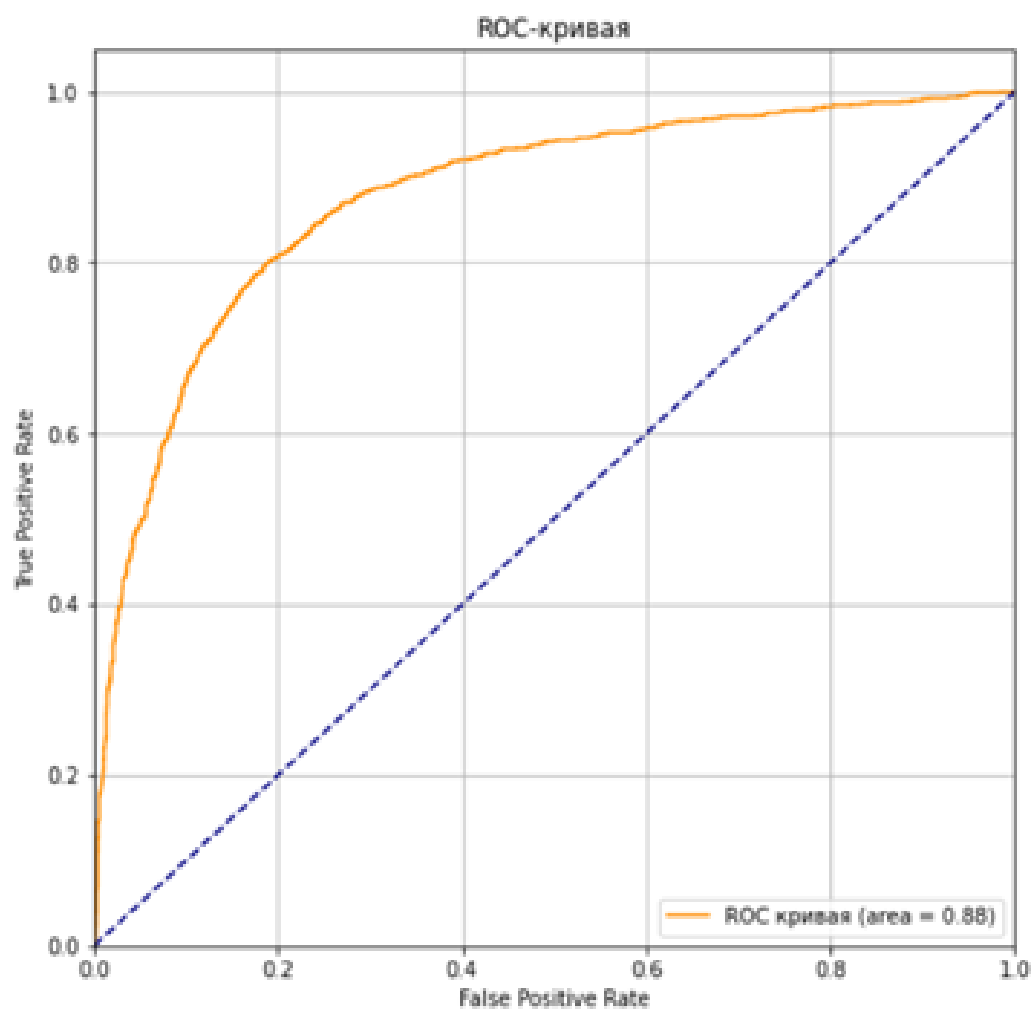


## Приложение 5.

## Показатели качества классификации kNN-классификатора

	Точность	Полнота	F1
Первый класс	0.81	0.87	0.84
Второй класс	0.76	0.66	0.7
Среднее по классам	0.78	0.77	0.77
Всзвешенное среднее по классам	0.79	0.79	0.79

ROC-кривая качества классификации LightGBM-классификатора





## Приложение 7.

## Показатели качества классификации LightGBM-классификатора

	Точность	Полнота	F1
Первый класс	0.79	0.93	0.85
Второй класс	0.83	0.58	0.68
Среднее по классам	0.81	0.75	0.77
Всзвешенное среднее по классам	0.8	0.8	0.79

## Приложение 8.

Значение функции потерь при обучении BERT-классификатора



## Приложение 9.

Показатели точности классификации BERT-классификатора

Выборка	Точность классификации
Валидационная	0.95
Отложенная	0.88