МИНОБРНАУКИ РОССИИ

Федеральное государственное автономное образовательное
учреждение высшего образования
«Южный федеральный университет»

Институт математики, механики
и компьютерных наук им. И. И. Воровича

Кафедра алгебры и дискретной математики

Северин Никита Николаевич

АВТОМАТИЧЕСКОЕ РЕФЕРИРОВАНИЕ ТЕКСТОВ С ПОМОЩЬЮ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
по направлению подготовки
02.03.02 – Фундаментальная информатика и информационные технологии

Научный руководитель –
ст. преподаватель, к.ф.-м.н. Юрушкин Михаил Викторович

Допущено к защите:
заведующий кафедрой _____ Штейнберг Б. Я.

Ростов-на-Дону – 2020

ОТЗЫВ

на выпускную квалификационную работу
студента 4 курса бакалавриата Северина Н. Н.
на тему «Автоматическое реферирование текстов с помощью методов машинного обучения»

Целью работы являлась разработка экстрактивной модели автоматического реферирования текстов русского языка для создания рефератов, имеющих признаки индикативного и информативного типов. Особенностью данной задачи является отсутствие открытых датасетов, в которых для каждого текста приведен его реферат или аннотация.

В рамках работы Северин Н.Н. исследовал подходы к автоматическому реферированию, не требующие специально размеченных корпусов текстов. Для этого был проведен сравнительный анализ существующих и разработка новых методов извлечения признаков из текстов, на основе которых были реализованы модификации графового алгоритма TextRank, а также алгоритмы, основанные на кластеризации предложений. Качество алгоритмов было оценено с помощью метрики ROUGE и методом экспертного оценивания. Проведенный анализ выявил особенности полученных алгоритмов и показал, что наилучшим является алгоритм, основанный на интеграции рассматриваемых подходов.

В процессе выполнения выпускной работы Северин Н.Н. проявил себя как подготовленный, вполне сформировавшийся специалист, способный к творческой и самостоятельной работе. Считаю, что представленная Севериным Н.Н. работа полностью соответствует требованиям, предъявляемым к выпускным квалификационным работам по направлению 02.03.02 «Фундаментальная информатика и информационные технологии» и заслуживает оценки «отлично».

Научный руководитель, ст. преп., к.ф.-м.н. М. В. Юрушкин

Задание на выпускную квалификационную работу студента 4-го года бакалавриата Северина Н. Н.

Направление подготовки: 02.03.02 — Фундаментальная информатика и информационные технологии.

Студент: Н. Н. Северин.

Научный руководитель: ст. преподаватель, к.ф.-м.н. М. В. Юрушкин.

Год защиты: 2020.

Тема работы: «Автоматическое реферирование текстов с помощью методов машинного обучения».

Цель работы: построение экстрактивной модели автоматического реферирования текстов русского языка для создания рефератов смешанного типа.

Задачи работы:

- Сравнительный анализ существующих подходов к задаче извлечения признаков из текста.
- Выбор метрик оценки качества работы алгоритмов реферирования текстов.
- Анализ подходов к задаче автоматического реферирования текстов и ее существующих решений.
- Разработка алгоритмов реферирования.
- Анализ и сравнение работы моделей.

Научный руководитель, ст. преп., к.ф.-м.н. М. В. Юрушкин

Студент 4-го курса бакалавриата Н. Н. Северин

25 ноября 2019 г.

АНТИПЛАГИАТ
ТВОРИТЕ СОБСТВЕННЫМ УМОМ

Южный Федеральный Университет

СПРАВКА
о результатах проверки текстового документа на наличие заимствований

Проверка выполнена в системе Антиплагиат.ВУЗ

Автор работы	**Северин Никита Николаевич**
Подразделение	ИММиКН, ФИИТ
Тип работы	Выпускная квалификационная работа
Название работы	Северин_Диплом (3)
Название файла	Северин_Диплом (3).pdf
Процент заимствования	**3.20 %**
Процент самоцитирования	**0.00 %**
Процент цитирования	**17.14 %**
Процент оригинальности	**79.66 %**
Дата проверки	**18:51:50 17 июня 2020г.**
Модули поиска	Модуль поиска ИПС "Адилет"; Модуль выделения библиографических записей; Сводная коллекция ЭБС; Модуль поиска "Интернет Плюс"; Коллекция РГБ; Цитирование; Модуль поиска переводных заимствований; Модуль поиска переводных заимствований по elibrary (EnRu); Модуль поиска переводных заимствований по интернет (EnRu); Коллекция eLIBRARY.RU; Коллекция ГАРАНТ; Коллекция Медицина; Диссертации и авторефераты НББ; Модуль поиска перефразирований eLIBRARY.RU; Модуль поиска перефразирований Интернет; Коллекция Патенты; Модуль поиска "ЮФУ"; Модуль поиска общеупотребительных выражений; Кольцо вузов
Работу проверил	Юрушкин Михаил Викторович
	ФИО проверяющего
Дата подписи	
	Подпись проверяющего

Чтобы убедиться в подлинности справки, используйте QR-код, который содержит ссылку на отчет.

Ответ на вопрос, является ли обнаруженное заимствование корректным, система оставляет на усмотрение проверяющего. Предоставленная информация не подлежит использованию в коммерческих целях.

Оглавление

Введение

Реферирование текста — это процесс выделения наиболее важной информации из текста для создания его сокращенной версии, исходя из конкретной цели [6]. Согласно исследованиям [2, 8] по полноте изложения информации можно выделить 3 вида рефератов: индикативный, информативный и смешанный.

Индикативный реферат указывает только на основные аспекты содержания первичного текста. Цель такого реферата — дать читателю представление о том, стоит ли обращаться к реферируемому документу вообще.

Информативный реферат, в отличие от индикативного, излагает основное содержание текста и служит источником сведений (информации) о фактах, представленных в исходном документе.

Смешанный реферат имеет признаки и информативного, и индикативного, и, как правило, используется при работе с объемным и сложным текстом, охватывающим несколько аспектов одной темы.

С точки зрения реализации существует два основных подхода к решению задачи автоматического реферирования текста:

1. *Абстрактное реферирование* (используются алгоритмы генерации для создания нового текста).

2. *Извлечение* (экстрактивное реферирование; реферирование на основе фраз и целых предложений, присутствующих в документе).

Данная работа посвящена разработке экстрактивной модели автоматического реферирования текстов русского языка для создания рефератов смешанного типа на основе предложений, входящих в исходный текст.

Постановка задачи

Целью данной работы является построение экстрактивной модели автоматического реферирования текстов русского языка для создания рефератов смешанного типа.

Для достижения поставленной цели в работе решаются следующие основные задачи:

- Сравнительный анализ существующих подходов к задаче извлечения признаков из текста.
- Выбор метрик оценки качества работы алгоритмов реферирования текстов.
- Анализ подходов к задаче автоматического реферирования текстов и ее существующих решений.
- Разработка алгоритмов реферирования.
- Анализ и сравнение работы моделей.

Основные понятия

В этом разделе вводятся основные понятия, которые используются в данной выпускной работе.

Обработка естественного языка (автоматическая обработка текстов, natural language processing, NLP) — междисциплинарное направление на стыке искусственного интеллекта и математической лингвистики, предметом изучения которого являются методы компьютерного анализа и синтеза естественных языков [7].

Лемматизация — это приведение слова к его начальной форме, которая называется *леммой*. Программы, выполняющие лемматизацию автоматически, называются *лемматизаторами*.

Стоп-слова — это часто употребляемые слова в текстах любого вида и тематики. Эмпирически выявлено, что их удаление из текста может существенно улучшить качество автоматической обработки текстов.

N-грамма — последовательность длины n.

Лексема — слово, рассматриваемое как единица словарного состава языка в совокупности его конкретных грамматических форм.

Токен — объект, создающийся из лексемы в процессе лексического анализа.

Корпус текстов — коллекция связанных между собой текстов, имеющая определенную разметку. Корпуса, как правило, оцифровываются и хранятся в электронном виде.

Датасет (для машинного обучения) — обработанная и структурированная информация в табличном виде.

1. Методы извлечения признаков из текста

Задача извлечения признаков из текста — общая задача выделения количественных закономерностей в тексте, которые могут быть использованы при решении других задач автоматической обработки текстов. Данная задача чаще всего решается при помощи построения языковых моделей текста.

Языковая модель текста — это вероятностное распределение на множестве словарных последовательностей [7]. Глобально существует 2 подхода к построению языковых моделей: модель мешка слов и дистрибутивная семантика.

1.1. Модель мешка слов (Bag of words)

Модель мешка слов основана на предположении о том, что для выявления закономерностей в тексте важна только частота употребления слов, но не их порядок. Данная модель может быть применена в случае, когда анализируемых текстов несколько, либо когда используется дополнительная коллекция документов (в зависимости от задачи документами можно считать в том числе и отдельные абзацы, предложения и словосочетания).

Модель работает следующим образом. Строится терм-документная матрица F, в которой столбцы соответствуют термам (ими могут быть слова, леммы, n-граммы слов и символов…), а строки — документам. При этом $F_{dt} = n_{dt}$, где n_{dt} — абсолютная частота употребления терма t в документе d. В результате каждый рассматриваемый текст (документ) представляется в виде вектора — соответствующей ему строки матрицы F.

Такой подход имеет несколько существенных недостатков:

1. Большая размерность и высокая разреженность получаемых представлений ($O(|V|)$, где $|V|$ — размер словаря всей коллекции).

2. Не учитываются языковые особенности употребления слов (например, то, что служебные части речи встречаются в текстах чаще знаменательной лексики).

Для понижения размерности получаемых векторов могут быть использованы различные матричные разложения и тематическое моделирование (SVD, LSA, LDA, ARTM…). Эти методы не исследуются в данной работе.

Другим способом получения более информативных репрезентаций текстов является взвешивание абсолютных частот в матрице F. Чаще всего функция взвешивания строится на основе меры TF-IDF, которая вычисляется следующим образом. Для каждого терма t и документа d подсчитываются:

1. Мера TF (term frequency): $TF(t,d) = \frac{n_{td}}{\sum_{t' \epsilon d} n_{t'd}}$.
2. Мера IDF (inverse document frequency): $IDF(t,d) = \log \frac{|D|}{|\{d \epsilon D | \ t \epsilon d\}|}$, где D — множество всех документов.
3. Итоговая мера $TF\text{-}IDF(t,d) = TF(t,d) \times IDF(t,d)$.

Согласно исследованиям (например, [32, 35]) алгоритмы, основанные на мере TF-IDF, хорошо зарекомендовали себя при решении многих практических задач. Данная функция взвешивания имеет ряд преимуществ:

- Интерпретируемость.
- Быстро вычисляется.
- В нее можно заложить знания о предметной области.

Однако, несмотря на все преимущества, TF-IDF строит вектора большой размерности и не учитывает семантику слов, что может быть критично при решении ряда задач.

Одним из подходов для моделирования значения в языке является дистрибутивная семантика. Она утверждает, что значение слова зависит от контекста, в котором оно употребляется. Языковые модели, являющиеся продуктами подходов, основанных на дистрибутивной семантике, глобально можно разделить на 3 типа:

- Счетные дистрибутивные языковые модели.
- Дистрибутивно-семантические языковые модели.
- Предиктивные языковые модели.

1.2. Счетные дистрибутивные модели

В традиционной дистрибутивной семантике каждая лексическая единица описывается вектором, где в качестве измерений или компонентов выступают другие слова лексикона, а в качестве значений этих компонентов — частота совместной встречаемости рассматриваемой единицы с этими словами (обычно взвешенная тем или иным образом) [9]. Полученная матрица называется матрицей совместной встречаемости слов. Существует большое количество различных способов взвешивания абсолютных частот, например, *коэффициент Дайса*, *log-likelihood* и др. [21]. Тем не менее, данный подход имеет те же недостатки, что и модель мешка слов: большая размерность получаемых представлений, высокая разреженность.

Для устранения этих недостатков используются нейросетевые модели, которые рассматриваются далее. При обучении этих моделей целевым представлением каждого слова (токена) является сжатый вектор относительно небольшого размера (далее — *embedding*).

1.3. Дистрибутивно-семантические языковые модели

При обучении дистрибутивно-семантических языковых моделей для вектора каждого токена максимизируется сходство с векторами соседей и минимизируется сходство с векторами токенов, его соседями не являющихся. Ограничением такого подхода является то, что для каждого слова формируется один и тот же вектор при употреблении его в различных значениях. Самыми распространенными представителями таких моделей являются *Word2vec*, *FastText* и *Glove* [12, 27, 30].

Ключевое различие между этими моделями заключается в том, что Word2vec и Glove рассматривают каждое слово как неделимое целое, в то время как FastText (который в некотором смысле является расширением Word2vec) рассматривает каждое слово как композицию n-грамм символов. Исследования [12, 20] показали, что это дает модели FastText важные преимущества при работе с флективными языками (коим является русский):

1. Строит более качественные векторные представления для редких слов.
2. Может строить вектора для слов, отсутствующих в обучающем корпусе.

1.4. Предиктивные языковые модели

Модели данного класса направлены на предсказание следующего слова по известному префиксу предложения (некоторые и по постфиксу), и, как правило, реализуются глубокими нейронными сетями. Позволяют работать с многозначными словами, строя их «контекстуализированные» вектора, и в настоящий момент показывают наилучшие результаты при решении большинства задач NLP. Самыми известными представителями таких моделей являются *BERT* (а также его модификация *RoBERTa*), *ELMo*, *GPT-2* [16, 31, 34].

В [17] приводится сравнение данных моделей на различных типах текстов, которое показывает, что в общем случае модель BERT строит наилучшие векторные репрезентации слов.

Модель BERT

BERT (Bidirectional Encoder Representations from Transformers) использует в своей основе архитектуру Transformer и механизм внимания (attention) [44], что позволяет ему анализировать длинные контекстные зависимости между словами в тексте. Архитектура Transformer состоит из двух отдельных механизмов: encoder, который читает входную

последовательность и создает для нее промежуточное представление, а также decoder, который по полученному представлению решает требуемую задачу. Для использования обученного BERT, как языковой модели, необходим только его encoder.

При обучении любой предиктивной модели важным шагом является выбор конечной цели обучения. Некоторые модели (например, GPT-2) учатся предсказывать слово по известному префиксу, что ограничивает возможность выявления длинных контекстных зависимостей в тексте. Для устранения этого недостатка во время обучения BERT решает следующие 2 задачи:

1) *Маскирование* (предсказание слова по его контексту слева и справа).

2) *Предсказание следующего предложения*. Модель получает пары предложений в качестве входных данных и учится предсказывать, является ли второе предложение следующим после первого в исходном тексте. Для того, чтобы модель могла отделить предложения друг от друга, в начало первого добавляется маркер [CLS], в конец каждого — маркер [SEP].

Кроме того, модель BERT может работать с «сырыми» текстами (не лемматизированными, без удаления пунктуации и т.п.) благодаря внутренней предобработке входных данных с помощью алгоритма byte pair encoding (BPE) [41], что значительно облегчает задачу подготовки датасетов для ее обучения.

В результате обучения модель BERT (как и многие другие предиктивные и дистрибутивно-семантические модели) может строить вектора для токенов входного текста. Далее для построения вектора всего текста в зависимости от решаемой задачи существуют различные стратегии агрегирования векторов токенов: выбрать вектор [CLS]-токена, усреднить вектора выходного слоя и др.

Модель Sentence-BERT

Несмотря на высокое качество решения задач обучения «с учителем», модель BERT не подходит для решения задач семантического

информационного поиска, кластеризации предложений и текстов [37]. В [37] предлагается модель Sentence-BERT (далее — SBERT) — модификация модели BERT, которая строит вектора для предложений текста, степень семантической близости которых может быть оценена косинусом угла между их векторами (косинусное расстояние).

Модель SBERT состоит из нескольких частей:

- Модель BERT.
- Операция усреднения векторов выходного слоя BERT.
- Сиамская сеть с TripletLoss (которая позволяет SBERT обучиться определению степени сходства входных предложений).

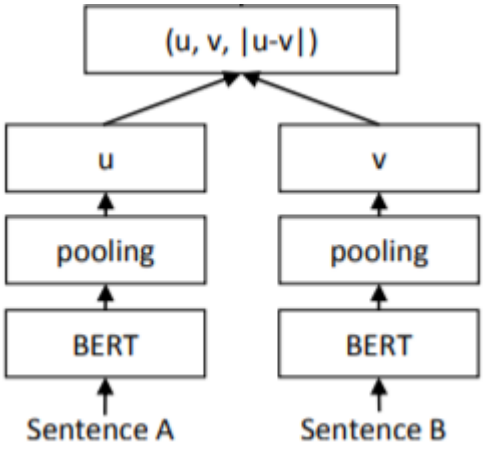

Рисунок 1.1. Принцип работы Sentence-BERT

1.5. Transfer learning в NLP

Использование глубоких нейронных сетей для создания языковой модели требует большого объема данных для сходимости обучения. Зачастую при решении частной задачи NLP необходимых корпусов текстов нужного размера может не существовать и создать языковую модель, основываясь только на этих данных, не представляется возможным. Transfer learning позволяет решить эту проблему.

Transfer learning (TL) — одно из направлений исследований в машинном обучении, которое изучает возможность применения знаний, полученных при

решении одной задачи, к другой. Основная идея TL в NLP заключается в использовании готовых моделей, которые обучались на огромных корпусах текстов, для извлечения базовых признаков из текстов частной задачи и дальнейшего их использования для ее решения.

Для русского языка готовые предобученные модели BERT, SBERT и FastText есть в библиотеке DeepPavlov [15].

2. Оценки качества алгоритмов реферирования текстов

Существует 2 подхода к оценке качества работы алгоритмов автоматического реферирования: автоматическое тестирование и экспертное оценивание.

Для проведения автоматического тестирования необходим набор текстов и их эталонных рефератов (т.е. написанных человеком). Самой распространенной количественной метрикой оценки качества алгоритмов реферирования на данный момент является метрика «ROUGE», которая согласно [6] имеет высокую корреляцию с человеческим оцениванием. Существует несколько модификаций данной метрики: «ROUGE-N», «ROUGE-L», «ROUGE-W», «ROUGE-S», «ROUGE-SU».

В данной работе используются метрики ROUGE-N, ROUGE-L, которые являются самыми распространенными на практике.

2.1. ROUGE-N

Данная метрика основана на вычислении количества n-грамм слов, которые встречаются как в эталонном реферате, так и в автоматически сгенерированном (далее — сгенерированном). Формула для ее вычисления имеет вид:

$$\text{ROUGE-N}(Gen, Ref) = \frac{\text{Count}_{\text{match}}(gram_n)}{\text{Count}_{\text{ref}}(gram_n)},$$

где $\text{Count}_{\text{match}}(gram_n)$ — количество n-грамм слов, содержащихся и в сгенерированном реферате Gen и в эталонном Ref; $\text{Count}_{\text{ref}}(gram_n)$ — количество n-грамм слов в эталонном реферате.

В данной работе используются метрики ROUGE-1 и ROUGE-2.

2.2. ROUGE-L

Определение 2.1. Последовательность $Z = [z_1, \dots, z_n]$ называется *подпоследовательностью* последовательности $X = [x_1, \dots, x_m]$, если $\exists [i_1, \dots, i_n], : \forall j = \overline{1..n} \Rightarrow z_j = x_{i_j}$ [6].

Пусть X — последовательность слов эталонного реферата длины n, Y — последовательность слов автоматического реферата длины m. Обозначим $LCS(X, Y)$ — длину наибольшей общей подпоследовательности между X и Y. Тогда формула для вычисления метрики ROUGE-L имеет вид:

$$\mathrm{P_{lcs}}(X, Y) = \frac{\mathrm{LCS}(X, Y)}{m}$$

$$\mathrm{R_{lcs}}(X, Y) = \frac{\mathrm{LCS}(X, Y)}{n}$$

$$\text{ROUGE-L}(Gen, Ref) = \frac{2\mathrm{P_{lcs}}(X, Y)\mathrm{R_{lcs}}(X, Y)}{\mathrm{P_{lcs}}(X, Y) + \mathrm{R_{lcs}}(X, Y)}$$

В отличие от ROUGE-N, данная метрика проверяет не совпадение подряд идущих слов двух текстов, а находит самую длинную общую подпоследовательность этих текстов, т.е. учитывает совпадение слов с возможными пропусками.

3. Обзор существующих решений задачи автоматического реферирования текстов

Самый большой прогресс в задаче автоматического реферирования достигнут сегодня для текстов на английском языке [23, 24, 28, 43]. Одна из причин — наличие открытых размеченных датесетов («BigPatent» [42], «CNN/DailyMail» [39] и др.), в которых для каждого текста приведен его реферат или аннотация.

Исследования показывают, что наилучшие результаты на данный момент показывают алгоритмы абстрактного реферирования [23] и алгоритмы, основанные на объединении абстрактного и экстрактивного подходов [43] (в обоих случаях с использованием методов глубокого обучения). Тем не менее, для создания таких моделей требуется большой объем размеченных данных, что делает их на сегодняшний день нереализуемыми для реферирования текстов русского языка (в силу отсутствия подходящих датасетов). Поэтому в данной работе рассматриваются методы автоматического реферирования, не требующие специальной разметки текстов, на основе экстрактивного подхода.

В данном разделе дана классификация методов экстрактивного реферирования и рассмотрены алгоритмы, использующиеся в экспериментальной части работы.

3.1. Классификация методов экстрактивного реферирования

На основе исследований [18, 29] можно дать следующую классификацию методов экстрактивного реферирования текстов:

- Выделение наиболее важных предложений из текста.
- Выделение ключевых фраз.
- Упрощение предложений текста.

В данной работе рассматриваются алгоритмы, основанные на выделении наиболее важных предложений из текста (далее — алгоритмы экстрактивного реферирования). Обобщенно такие алгоритмы состоят из следующих этапов:

1. Предварительная обработка исходного текста (может включать лемматизацию, удаление знаков препинания, стоп-слов и др.).
2. Извлечение признаков из текста.
3. Выделение наиболее важных предложений из текста и построение итогового реферата.

Согласно [4] существует 5 типов методов для реализации шагов 2, 3:

- Статистические методы.
- Графовые методы.
- Методы на основе машинного обучения.
- Методы на основе построения семантических связей.
- Смешанные подходы.

В [4] было проведено исследование, сравнивающее 3 алгоритма реферирования:

1. TextRank (графовый алгоритм, показывающий наилучшее качество среди других графовых подходов) [26].
2. Кластеризация предложений с помощью K-Means (выделение признаков с помощью TF-IDF) [33].
3. Латентно-семантический анализ (LSA).

По его результатам, было показано, что наилучшее качество на текстах русского языка показывает классический алгоритм TextRank.

В данной выпускной работе исследуются способы улучшения алгоритмов 1, 2 с помощью различных стратегий извлечения признаков из текста методами машинного обучения.

3.2. Описание используемых алгоритмов

3.2.1. TextRank

Данный алгоритм является представителем графовых подходов. Его применение к задаче автоматического реферирования текстов представлено в работе [26]. Алгоритм заключается в следующем:

1. На основе текста строится взвешенный неориентированный граф, вершины которого соответствуют предложениям текста. Весом ребра между двумя вершинами является степень схожести двух предложений, соответствующих вершинам. В классическом алгоритме TextRank она вычисляется как количество совпадающих слов в предложениях, нормированное суммарной длиной этих предложений.

2. Исходя из весов ребер, с помощью алгоритма pagerank [11] в итерационном процессе каждой вершине присваивается вес по следующей формуле:

$$W(V_i) = (1-d) + d \times \sum_{V_j \in Inc(V_i)} \frac{w_{ij}}{\sum_{V_k \in Inc(V_j)} w_{jk}} W(V_j)$$

где V_i, V_j — вершины графа;

$Inc(V_i)$ — множество вершин, смежных с вершиной V_i;

w_{ij} — вес ребра между вершинами V_i, V_j;

d — коэффициент затухания, в данном алгоритме равен 0.85.

Итерационный процесс завершается, как только веса вершин перестают меняться более, чем на 0.0001.

3. После вычисления своих весов вершины упорядочиваются по убыванию значения веса и в реферат включаются предложения, соответствующие первым n вершинам, где n — требуемое количество предложений в реферате.

3.2.2. Кластеризация предложений

Применение кластеризации предложений к задаче автоматического реферирования было предложено в работе [33]. Алгоритм состоит из следующих шагов:

1. Входной текст представляется в виде совокупности его предложений, из каждого предложения извлекаются признаки.
2. Исходя из выделенных признаков, производится кластеризация предложений.
3. Итоговый реферат формируется из предложений, наиболее близких к центроидам полученных кластеров.

В [39] предлагается использовать K-Means в качестве алгоритма кластеризации, т.к. он позволяет автоматически определить центроиды кластеров, а также работать с любым количеством предложений за приемлемое время (в отличие от, например, спектральной кластеризации). В силу этого в данной работе используется именно этот алгоритм кластеризации.

Для определения оптимального числа кластеров в работе используется оценка «Силуэт» (Silhouette Coefficient) [22]: выбирается число кластеров, максимизирующее данную оценку.

4. Эксперименты

4.1. Описание тестового набора данных

Тестовый датасет представляет собой набор 4200 научных текстов, разделенных на 3 категории: естественнонаучные и социально-экономические тексты, гуманитарные тексты, технические тексты. Данные были автоматически собраны с сайта онлайн-издательства «NotaBene» [5], из каждой статьи были выделены (рис 4.1.):

1. Название.
2. Аннотация.
3. Ключевые слова.
4. Текст статьи.

title	annotation	key_words	text
Отражение сибирской идентичности в искусстве ...	Объектом исследования в данной статье является...	сибирская идентичность, этнос, архаичность, ис...	Во-первых, сибирская идентичность базируется н...

Рисунок 4.1. Пример одной записи из набора данных

Статистика по полученному набору данных представлена на рис. 4.2 и в таблице 4.1.

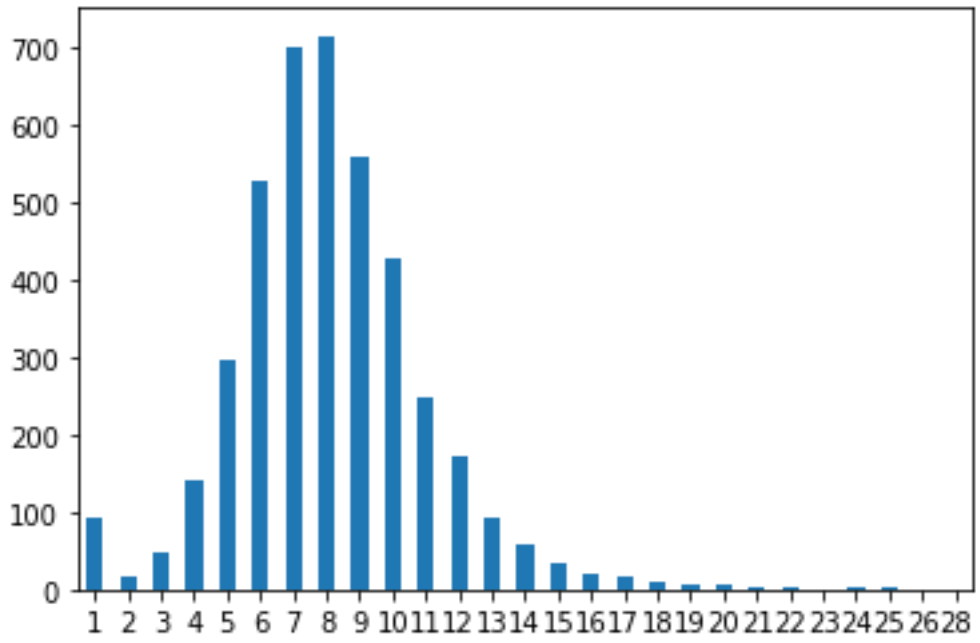

Рисунок 4.2. Распределение количества предложений в аннотациях

Характеристика	Mean	Min	Max
Количество предложений в аннотациях	8	1	28
Количество предложений в статьях	77	21	5708

Таблица 4.1. Статистика количества предложений

Особенности используемого датасета:

– Эталонные рефераты (аннотации) имеют небольшую длину: в основном 6–9 предложений, в то время как исходные статьи имеют среднюю длину 76–77 предложений.

– Набор исходных текстов разнообразен: в нём присутствуют и совсем короткие тексты (длиной 21 предложение), и достаточно длинные (длиной более 5000 предложений).

– Эталонные рефераты не являются экстрактивными, т.е. предложения в них переформулированы, некоторые слова и фразы могут не встречаться в исходных текстах.

4.2. Разработка алгоритмов реферирования

За основу были взяты алгоритмы, описанные в 3.2. Для извлечения признаков из текстов исследовались модели:

1. *Модель мешка слов + TF-IDF*. В рамках данной модели каждое предложение входного текста рассматривается как самостоятельный документ. Для корректной работы все предложения предварительно лемматизировались, и из них удалялась пунктуация.

2. *FastText* (предобученная модель из библиотеки DeepPavlov [38]). Вектор предложения строится нормализацией и последующим усреднением векторов токенов, входящих в него, как описано в [12]. Рассматривались 2 варианта применения данной модели: с предварительным удалением знаков препинания и без него.

3. *RuBERT* (сокращенное название BERT для русского языка, предобученная модель из библиотеки DeepPavlov [38]). В [28] предлагается алгоритм экстрактивного реферирования текстов английского языка на основе

кластеризации K-Means и извлечения признаков из текста с помощью модели BERT. В статье показывается, что наилучшей стратегией получения вектора предложения моделью является усреднение векторов ее выходного слоя. При работе с RuBERT в данной работе использовался этот же подход. Рассматривались 2 варианта применения данной модели: с добавлением к предложениям маркеров [CLS] и [SEP] и без них.

4. *RuSBERT* (сокращенное название Sentence-BERT для русского языка, предобученная модель из библиотеки DeepPavlov [38]). Данная модель обучалась на тех же текстах, что и Sentence-BERT для английского языка (далее — EnSBERT), автоматически переведенных на русский язык. Рассматривались 2 варианта применения данной модели: с добавлением к предложениям маркеров [CLS] и [SEP] и без них.

5. *CustomSBERT* (самостоятельно обученная модель для построения векторов предложений русского языка). В работе данная модель сокращенно обозначается CSBERT.

При работе со всеми моделями для разделения текста на предложения использовалась библиотека rusenttokenizer [10]. Все описываемые далее алгоритмы были реализованы на ЯП Python.

Далее в текущем разделе более подробно рассматривается модель CustomSBERT, и дается описание итоговых моделей реферирования.

4.2.1. Модель CustomSBERT

Создание данной модели обусловлено предположением, что, т.к. модель RuSBERT обучена на текстах, полученных машинным переводом, она может работать не столь качественно (для текстов русского языка), как модель EnSBERT (для текстов английского языка). Модель CustomSBERT была разработана на основе подхода, описанного в [36], который опирается на гипотезу, что одни и те же предложения на разных языках должны иметь схожие векторные представления. Данный подход заключается в следующем.

Пусть требуется построить модель $\bar{M}$, которая отображает множество предложений естественного языка L в некоторое векторное пространство. Предположим, что такая модель M существует для другого естественного языка T (назовем ее моделью-учителем). Кроме того, предположим, что существует параллельный корпус предложений для языков T, L, т.е. набор предложений $\{(t_i, l_i) \mid t_i \in T, l_i \in L\}$. Тогда обучение модели $\bar{M}$ может производиться таким образом, чтобы для входных предложений $t_i \in T, l_i \in L$ из параллельного корпуса модель $\bar{M}$ строила такие векторные представления, чтобы $\bar{M}(t_i) \approx M(t_i)$, $\bar{M}(l_i) \approx M(t_i)$. Более формально: для входного мини-батча $B = \{(t_i, l_i) \mid t_i \in T, l_i \in L\}$ минимизируется среднеквадратичная функция потерь (mean-squared loss, MSE-loss):

$$\frac{1}{|B|}\sum_{i=1}^{|B|}\left[\left(M(t_i) - \bar{M}(t_i)\right)^2 + \left(M(t_i) - \bar{M}(l_i)\right)^2\right]$$

В качестве модели-учителя для CustomSBERT использовалась модель EnSBERT, в качестве базовой архитектуры — архитектура Sentence-BERT, в основе которой лежит предобученная XLM-RoBERTa [14]. Выбор данной архитектуры обусловлен тем, что XLM-RoBERTa обучалась на мультиязычном корпусе, включающем тексты русского языка и, значит, может одновременно работать с русским и английским языками.

Материалом для обучения был англо-русский параллельный корпус (1 млн параллельных предложений) [1] и часть датасета [40], вручную переведенная на русский язык, которая использовалась для оценивания способности обучаемой модели предсказывать степень близости предложений, как это делалось при обучении EnSBERT [37].

Модель обучалась со следующими параметрами:

- *Размер мини-батча* = 8.
- *Количество эпох* = 16.
- *Метод оптимизации*: Adam с *learning rate* = 2e-5.

Схема обучения CustomSBERT представлена на рис.4.3.

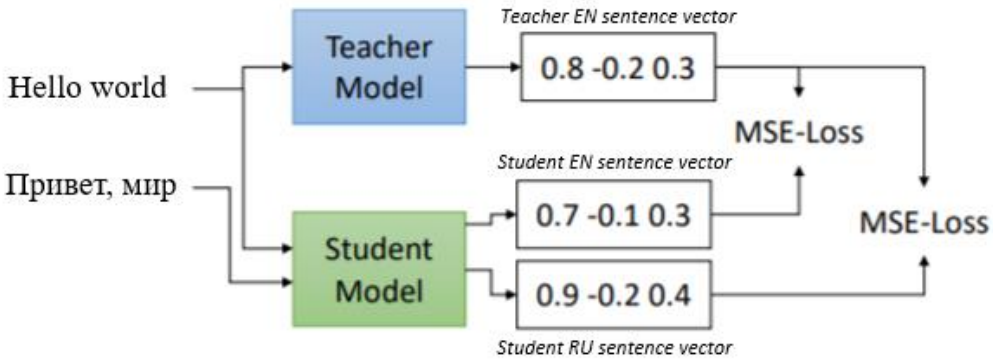

Рисунок 4.3. Схема обучения CustomSBERT

4.2.2. Итоговые алгоритмы реферирования

Для всех описываемых ниже алгоритмов анализировались 2 стратегии предобработки текста: с сохранением всех предложений текста и с предварительным удалением коротких предложений (на основе предположения, что такие предложения не должны входить в итоговый реферат). Под короткими предложениями в данной работе понимаются предложения, содержащие менее 20 символов.

1. **<u>Алгоритмы на основе TextRank</u>**

Для применения моделей 2–5 из 4.2. к извлечению признаков для их дальнейшего использования алгоритмом PageRank (3.2.1) был модифицирован способ оценивания схожести двух предложений. Вместо подсчета количества совпадающих слов близость предложений оценивалась косинусным расстоянием между их векторами (полученными моделями 2–5).

Кроме того, исследовался еще один способ подсчета схожести (на основе модели мешка слов). Пусть s_1, s_2 — сравниваемые предложения, состоящие из набора слов $(x_1, \dots x_n)$ и $(y_1, \dots y_m)$ соответственно. Для каждого предложения строится вектор размерности $|\{x_1, \dots x_n, y_1, \dots y_m\}|$, в котором компонента, соответствующая слову w, равна количеству вхождений w в соответствующее предложение. Степень схожести s_1 и s_2 определяется как косинусное расстояние между полученными векторами. Алгоритм на основе объединения данного подхода и PageRank далее будет обозначаться

Simple_PRank. Для корректной работы данного алгоритма предварительно проводилась лемматизация входного текста и удаление из него стоп-слов.

Использование модели 1 из 4.2. и дальнейшее применение алгоритма PageRank не исследовались, поскольку в [4] было показано, что полученный алгоритм будет работать хуже по качеству, чем классический алгоритм TextRank.

В результате были разработаны следующие алгоритмы реферирования:

- CSBERT_Prank (сокр. название для CustomSBERT + PageRank)
- FT_Prank-Clean (сокр. название для FastText + PageRank с предварительным удалением знаков препинания из текста)
- FT_Prank-Raw (сокр. название для FastText + PageRank, работающим с исходным текстом).
- RuBERT_PRank (сокр. название для RuBERT + PageRank без использования специальных маркеров [CLS], [SEP]).
- RuBERT_PRank-ST (сокр. название для RuBERT + PageRank с использованием специальных маркеров [CLS], [SEP]).
- RuSBERT_PRank (сокр. название для RuSBERT + PageRank без использования специальных маркеров [CLS], [SEP]).
- RuSBERT_PRank-ST (сокр. название для RuSBERT + PageRank с использованием специальных маркеров [CLS], [SEP]).
- Simple_Prank (описан выше).
- TextRank (классический алгоритм TextRank).

2. <u>Алгоритмы на основе кластеризации K-Means</u>

Для построения алгоритмов на основе кластеризации K-Means был использован алгоритм, описанный в 3.2.2. В результате были получены следующие алгоритмы реферирования:

- CSBERT_KMeans (сокр. название для CustomSBERT + K-Means)
- FT_KMeans-Clean (сокр. название для FastText + K-Means с предварительным удалением знаков препинания из текста)

– FT_KMeans-Raw (сокр. название для FastText + K-Means, работающим с исходным текстом).
– RuBERT_KMeans (сокр. название для RuBERT + K-Means без использования специальных маркеров [CLS], [SEP]).
– RuBERT_KMeans-ST (сокр. название для RuBERT + K-Means с использованием специальных маркеров [CLS], [SEP]).
– RuSBERT_KMeans (сокр. название для RuSBERT + K-Means без использования специальных маркеров [CLS], [SEP]).
– RuSBERT_KMeans-ST (сокр. название для RuSBERT + K-Means с использованием специальных маркеров [CLS], [SEP]).
– TF-IDF_KMeans (сокр. название для TF-IDF (модель 1 из 4.2.) + K-Means).

4.3. Анализ работы алгоритмов и их сравнение

Оценивание алгоритмов было проведено автоматически и вручную.

4.3.1. Автоматическое оценивание

Автоматическое оценивание проводилось по метрикам из 2.1. и 2.2., а также подчитывалась доля ключевых слов, которые вошли в сгенерированный реферат.

Результаты автоматического оценивания алгоритмов представлены в таблице 4.2. В ней приведены наивысшие оценки алгоритмов, которые были получены при использовании различных стратегий их применения и предобработки текстов (4.2., 4.2.2.).

Алгоритм\Оценка	ROUGE-1	ROUGE-2	ROUGE-L	Key words
CSBERT_KMeans	0.2732	0.0724	0.1749	0.6438
CSBERT_PRank	0.1709	0.0267	0.1218	0.4815
FT_KMeans	0.2747	0.0714	0.1783	0.6344
FT_PRank	0.1004	0.0202	0.0848	0.4656
RuBERT_KMeans	0.2793	0.0742	0.1815	0.6400
RuBERT_PRank	0.1422	0.0471	0.1169	0.5227
RuSBERT_KMeans	0.2632	0.0674	0.1715	0.6278
RuSBERT_PRank	0.1932	0.0691	0.1526	0.5900
Simple_PRank	*0.3075*	*0.0833*	*0.1885*	*0.6947*
TextRank	**0.3436**	**0.1153**	**0.2157**	**0.7159**
TF-IDF_KMeans	*0.2936*	*0.0816*	*0.1878*	*0.6890*

Таблица 4.2. Качество работы алгоритмов, автоматическое оценивание

Согласно данным оценкам наилучшими моделями реферирования являются TextRank, SimplePageRank и TF-IDF_KMeans. Тем не менее, в исследовании [43] было показано, что автоматические оценки работы алгоритмов реферирования на научных текстах имеют серьезные ограничения и сравнение алгоритмов только на их основании не может быть валидным.

4.3.2. Метод экспертного оценивания

Для проведения оценивания вручную из 3-х категорий текстов, описанных в 4.1., случайным образом были выбраны 10 текстов, для которых по шкале от 0 до 5 оценивалось качество работы исследуемых алгоритмов. Критериями для оценивания являлись:

- Отражение основных смысловых частей исходного текста.
- Логическая согласованность и возможность для читателя понять, о чем говорится в исходном тексте.

Оценивание производилось четырьмя независимыми респондентами, итоговые оценки формировались усреднением полученных результатов каждым респондентом. Далее для каждого алгоритма оценки за все категории текстов суммировались.

Итоговые результаты представлены на рис. 4.4.

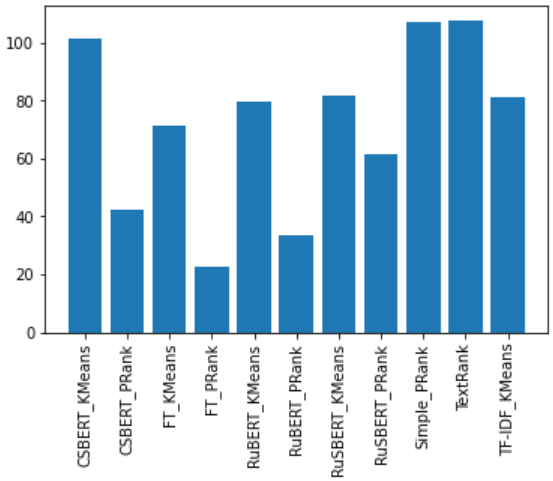

Рисунок 4.4. Качество работы алгоритмов, экспертное оценивание

Более подробные таблицы приведены в приложении А.

Пример оценивания результатов работы двух алгоритмов

Алгоритм 1.

Задача раннего обнаружения решается при наземном мониторинге, который дает возможность осуществлять непрерывный контроль, за лесными территориями и выявлять пожары на ранней стадии. Основная научно-техническая задача состоит в создании системы распределенного видеонаблюдения для решения задачи раннего обнаружения лесных пожаров. В работе проанализированы существующие подходы в области мониторинга лесных пожаров. Предлагаемая разработка находится на стадии проектирования. Работа выполнена при финансовой поддержке РФФИ, проект 13-01-00427а.

Алгоритм 2.

Задача раннего обнаружения решается при наземном мониторинге, который дает возможность осуществлять непрерывный контроль, за лесными территориями и выявлять пожары на ранней стадии. Основная научно-техническая задача состоит в создании системы распределенного видеонаблюдения для решения задачи раннего обнаружения лесных пожаров. Существенными недостатками используемых в настоящее время методов обнаружения является невозможность раннего обнаружения, автоматизации процесса обнаружения и определения местоположения очага возгорания[7]. Проектируемая система предназначена для обнаружения лесных пожаров и определения их пространственных координат, в масштабе реального времени. Она обеспечивает высокую эффективность обнаружения лесных пожаров и определения их координат. В качестве основного аппаратного модуля для визуального наблюдения будет использоваться видеокамера Axis Q6032-E[8].Купольная IP-камера с механически реализованными функциями PTZ и 420-кратным общим масштабированием (рис.2) предназначена для контроля обширных многолюдных пространств. Максимальное увеличение (оптическое + цифровое).

В данном примере первый реферат является более логически согласованным, но при этом не позволяет определить, насколько подробно описывается разрабатываемая система (в общих чертах или с точки зрения реализации?). Он является только индикативным.

Второй реферат, напротив, дает некоторую информацию о разрабатываемой системе и позволяет понять, что в статье описываются детали реализации. В то же время он является достаточно логически согласованным. В силу этого, данный реферат получил более высокие оценки, чем первый реферат.

4.3.3. Анализ полученных результатов

По результатам анализа работы алгоритмов были выявлены следующие закономерности:

1. Алгоритмы лучше всего работают на «гуманитарных» текстах. Это может быть связано с тем, что в анализируемых статьях этой категории информация представлена, в основном, в виде текста (нет формул, мало графических объектов).
2. Удаление коротких предложений повысило качество работы большинства моделей.
3. Извлечение признаков с помощью рассматриваемых нейросетевых моделей не подходит для дальнейшего их использования в алгоритме PageRank.
4. Лучшими среди анализируемых графовых алгоритмов являются классический TextRank и Simple_PRank.
5. Лучшими среди алгоритмов, основанных на кластеризации предложений, являются CSBERT_KMeans, RuSBERT_KMeans-ST, TF-IDF_KMeans, RuBERT_KMeans-ST. Следовательно, для извлечения признаков из предложений и их дальнейшей кластеризации в рамках реферирования текста лучше всего подходят модели архитектуры Sentence-BERT.

6. По результатам проведенного вручную оценивания наилучшими алгоритмами реферирования являются TextRank (107.5), Simple_ PRank (107) и CSBERT_KMeans (101.5).

Более детальный анализ показал, что алгоритмы TextRank, Simple_PRank работают схоже и выделяют в тексте самую ярко выраженную смысловую часть. Благодаря этому рефераты, получаемые этими моделями, являются наиболее логически согласованными.

Модель CSBERT_KMeans, напротив, делит текст на несколько смысловых частей и выбирает наиболее репрезентативные предложения из каждой.

Тем не менее, указанные особенности не позволяют моделям качественно справляться со сложными текстами, в которых присутствует несколько подробно описываемых смысловых частей. Это проиллюстрировано в следующем примере.

Пример (статья «Разработка методов адаптации пользовательских интерфейсов для USSD-сервисов»).

Эталонный реферат.

Рост числа электронных устройств и их возможностей провоцирует развитие рекомендательных систем в целом и адаптивных интерфейсов в частности. Такие системы собирают информацию о пользовательском опыте и при помощи методов машинного обучения анализируют полученные данные. Растет не только количество устройств, но и пользователей, которые часто имеют различные нужды и требования. Особое внимание в данной статье уделяется интерфейсу USSD-сервисов. Целью работы является разработка подхода, позволяющего снизить среднее количество запросов, совершаемых пользователями, в процессе работы с текстовым меню. Для поиска подходящего решения были применены методы системного анализа, в результате была установлена, сильная зависимость между потребностями каждого пользователя и контексте, в которой он находится. Использование информации о контексте выполнения, позволило улучшить существующие методы и получить высокие результаты. В основе подхода лежит использование двух этапной кластеризации: кластеризация по интересам на первом этапе и по контекстам – на втором. Для тестирования были использованы данные, собранные в процессе взаимодействия пользователей с USSD-сервисом.

TextRank.

В то же время число абонентов огромно, поэтому не представляется возможным хранение индивидуального варианта меню для каждого пользователя. Группировка пользователей в кластеры позволяет оптимизировать интерфейс для каждой конкретной группы, при этом избежав больших накладных расходов для хранения индивидуального дерева меню. В данной статье предлагается использовать информацию о контексте (или окружении) пользователя. Для каждой группы происходит выборка всех сессий, которые были открыты пользователями данного кластера. Где K – количество пользователей кластера, `n_(j)` — количество запросов, совершаемых пользователем в оптимизированном меню, `m_(j)` — количество запросов, совершаемых пользователем в первоначальном меню. Также был разработан метод, позволяющий адаптировать меню на основе данных, собранных в процессе работы пользователей с сервисом.

Simple_PRank.

Данная технология пользуется большим спросом в странах, где мобильный интернет слабо развит. Дизайн пользовательского интерфейса влияет на количество усилий, прилагаемых пользователем для достижения своих целей, а также на скорость освоения программной системы. Простейшим примером таких систем может служить браузер, который выносит наиболее часто посещаемые сайты на главную страницу или дополняет адрес при вводе первых символов. Для открытия сессии пользователь набирает команду на телефоне вида *.#. В процессе изучения литературы посвященной вопросу создания адаптивных интерфейсов, бала изучена статья [1], в которой предлагалось разделить модель пользователя на 3 части: информация о пользовательском опыте, информация об устройстве, информация о контексте. Тестирование показало, что разработанный метод, использующий двухэтапную кластеризацию, позволяет сократить среднее количество действий пользователей на 34%.

CSBERT_KMeans.

Наиболее важными из них являются: точность предлагаемых изменений относительно реакции пользователя, степень изменения и предсказуемости интерфейса, частота выполняемых адаптаций. Правильный выбор модели является крайне важным этапом разработки алгоритма, так как хранение лишней информации ресурсозатратно, а отсутствие какой-либо информации чревато потерей времени. Поскольку данная работа посвящена разработке метода автоматической адаптации USSD-меню, стоит немного рассказать о данной технологии. На вход нейронной сети подаются данные о кластеризуемом объекте. Этого должно хватить, поскольку пользователи открывшие данные контексты имеют схожие группы интересов, следовательно, имеют схожие контексты выполнения. Этот процесс выполняется для каждого кластера полученного на первом этапе (см. рис. 2). Каждой вершине присвоим вес `w_(i)` , определим это число следующим образом.

В анализируемой статье есть 2 основные темы: интерфейс USSD-сервисов и использование кластеризации пользователей для снижения количества запросов к текстовому меню.

В данном примере алгоритм TextRank выделил тему «кластеризация»; Simple_PRank преимущественно выделил тему «интерфейс» (при этом про USSD-сервисы не было сказано) и немного затронул тему «кластеризация»; CSBERT выделил обе темы, однако в силу логической несогласованности понять это крайне сложно.

4.4. Объединение алгоритмов TextRank и K-Means

По результатам сравнения алгоритмов были выявлены недостатки обоих подходов, а также показано, что они дополняют друг друга. Это сделало актуальным создание алгоритма, использующего оба подхода.

На основе TextRank и CSBERT (алгоритмов, показывающих наилучшие результаты) был разработан алгоритм «Mixed», который заключается в том, что сначала текст реферируется алгоритмом CSBERT, а затем в полученный реферат добавляются первые n предложений, которые TextRank определил как наиболее релевантные (n зависит от длины исходного текста и от длины реферата, полученного CSBERT).

Результаты оценивания данного алгоритма и его сравнение с другими приведено в таблице 4.3. и на рисунке 4.5.

Алгоритм\Оценка	ROUGE-1	ROUGE-2	ROUGE-L	Key words
Mixed	0.3383	0.1098	0.2079	0.7194
TextRank	0.3436	0.1153	0.2157	0.7159

Таблица 4.3. Качество работы алгоритмов, автоматическое оценивание

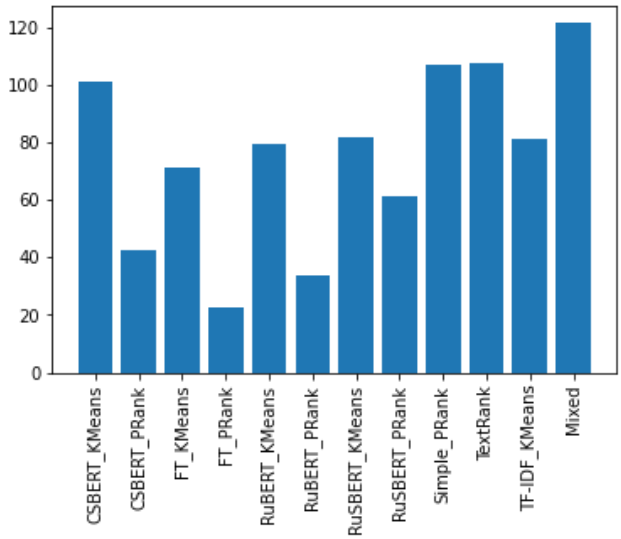

Рисунок 4.5. Качество работы алгоритмов, экспертное оценивание

Пример работы алгоритма (текст из 4.3.3).

В данной статье предлагается использовать информацию о контексте (или окружении) пользователя. Наиболее важными из них являются: точность предлагаемых изменений относительно реакции пользователя, степень изменения и предсказуемости интерфейса, частота выполняемых адаптаций. Каждый алгоритм адаптации так или иначе использует математическую модель пользователя. Правильный выбор модели является крайне важным этапом разработки алгоритма, так как хранение лишней информации ресурсозатратно, а отсутствие какой-либо информации чревато потерей времени. Поскольку данная работа посвящена разработке метода автоматической адаптации USSD-меню, стоит немного рассказать о данной технологии. На вход нейронной сети подаются данные о кластеризуемом объекте. Этого должно хватить, поскольку пользователи открывшие данные контексты имеют схожие группы интересов, следовательно, имеют схожие контексты выполнения. Этот процесс выполняется для каждого кластера полученного на первом этапе (см. рис. 2). Каждой вершине присвоим вес `w_(i)` , определим это число следующим образом. Также был разработан метод, позволяющий адаптировать меню на основе данных, собранных в процессе работы пользователей с сервисом.

В данном примере алгоритм выделил обе темы «интерфейс USSD-сервиса» и «кластеризация» и построил более логически согласованный реферат, чем CSBERT.

Таким образом, было показано, что алгоритм Mixed показывает наилучшее качество в задаче экстрактивного реферирования текстов русского языка среди других рассматриваемых алгоритмов, что означает, что интеграция графовых подходов и подхода кластеризации является перспективным направлением для исследования сегодня.

Заключение

В данной работе были изучены существующие методы извлечения признаков из текста и построена новая модель для русского языка. На основе этих методов, алгоритмов TextRank и кластеризации K-Means на ЯП Python были разработаны алгоритмы реферирования текстов русского языка. Исходный код доступен в GitHub-репозитории.

Были проведены эксперименты, в ходе которых на наборе научных статей данные алгоритмы были автоматически оценены по четырем метрикам, а также оценены вручную. Анализ работы алгоритмов показал, что лучший алгоритм, основанный на TextRank, и лучший алгоритм, основанный на кластеризации K-Means, дополняют друг друга (один выделяет наиболее выраженную тему в тексте, второй — основные смысловые части текста).

В итоге на основе объединения этих алгоритмов был разработан алгоритм «Mixed», который показал наилучшее качество работы с точки зрения и автоматического оценивания, и оценивания вручную.

Таким образом, данное исследование показало, что интеграция графовых подходов и подходов, основанных на кластеризации, является перспективным направлением для исследования в контексте решения задачи экстрактивного реферирования текстов русского языка.

Список литературы

1. Англо-русский параллельный корпус [Электронный ресурс] — URL: https://translate.yandex.ru/corpus.
2. Басова И.А. *Характеристика и типология рефератов как продуктов реферирования*. / И. А. Басова // Развитие образования. — 2018. — №1(1). — С. 78-81.
3. Ганский П. Н. *Интернет-пространство как особая коммуникационная среда и его влияние на современные общества*. / П.Н. Ганский // Теория и практика общественного развития. — 2015. — №17. — С. 118-121.
4. Новосёлова М.А. *Исследование методов автоматического реферирования текстов*: ВКР. — Спб., 2017.
5. Онлайн-издательство «NotaBene» [Электронный ресурс] — URL: https://www.nbpublish.com.
6. Павлович Б. *Исследование и реализация методов многоязыкового автоматического реферирования текстов*: ВКР. — М., 2012.
7. Профессиональный информационно-аналитический ресурс, посвященный машинному обучению, распознаванию образов и интеллектуальному анализу данных. [Электронный ресурс] — URL: http://www.machinelearning.ru/.
8. Соловьёв В.И. *Реферирование в научно-информационной деятельности и критерий редакторской оценки его содержания и формы*: Дис. ... канд. филол. наук. — М., 1970.
9. Школа лингвистики ВШЭ [Электронный ресурс] — URL: https://ling.hse.ru/embeddings.
10. A simple and fast rule-based sentence segmentation [Электронный ресурс] — URL: https://github.com/deepmipt/ru_sentence_tokenizer.

11. Bar-Yossef, Z., and Mashiach, L.-T. *Local approximation of pagerank and reverse pagerank*. // In Proceedings of the 17th ACM conference on Information and knowledge management (2008), ACM, pp. 279-288.

12. Bojanowski P. *Enriching Word Vectors with Subword Information* / P. Bojanowski, E. Grave, A. Joulin, T. Mikolov // arXiv.org. — 2016. [Электронный ресурс] — URL: https://arxiv.org/abs/1607.04606.

13. C.-Y. Lin. ROUGE: *A package for automatic evaluation of summaries*// Proceedings of ACL Text Summarization Branches Out Workshop. — 2004. — pp. 74–81.

14. Conneau A. *Unsupervised Cross-lingual Representation Learning at Scale* / A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov // arXiv.org. — 2019. [Электронный ресурс] — URL: https://arxiv.org/abs/1911.02116.

15. Deeppavlov: an open source library for deep learning end-to-end dialog systems and chatbots. [Электронный ресурс] — URL: https://github.com/deepmipt/DeepPavlov.

16. Devlin J. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding* / J. Devlin, M. W. Chang, K. Lee, K. Toutanova // arXiv.org. — 2019. [Электронный ресурс] — URL: https://arxiv.org/abs/1810.04805.

17. Ethayarajh K. *How Contextual are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings* / K. Ethayarajh // arXiv.org. — 2019. [Электронный ресурс] — URL: https://arxiv.org/abs/1909.00512.

18. Gambhir M. *Recent automatic text summarization techniques: a survey* / M. Gambhir, V. Gupta // Artificial Intelligence Review. — 2016. — pp. 1-66.

19. Hochreiter S. *Long short-term memory* / S. Hochreiter, J. Schmidhuber // Neural Computation — 1997. — № 9(8) — pp. 1735–1780.

20. Joulin A. *Bag of Tricks for Efficient Text Classification* / A. Joulin, E. Grave, P. Bojanowski, T. Mikolov // arXiv.org. — 2016. [Электронный ресурс] — URL: https://arxiv.org/abs/1607.01759.

21. Jurafsky D., Martin. J. H. *Speech and Language Processing*. Prentice Hall, 2019. — 613 pp.

22. Layton R. *Evaluating authorship distance methods using the positive Silhouette coefficient*. / R. Layton, P. Watters, R. Dazeley // Natural Language Engineering — 2013. — № 19(4) — pp. 517-535.

23. Lewis M. *Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension*. / M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, L. Zettlemoyer // arXiv.org. — 2019. [Электронный ресурс] — URL: https://arxiv.org/abs/1910.13461.

24. Liu Y. *Fine-tune BERT for extractive summarization* / Y. Liu // arXiv.org. — 2019. [Электронный ресурс] — URL: https://arxiv.org/abs/1903.10318.

25. Manning C. D., Raghavan P., Schütze H. *Introduction to Information Retrieval*. Cambridge University Press, 2009. — 544 с.

26. Mihalcea R. *TextRank: Bringing Order into Texts*. / R. Mihalcea, P. Tarau // Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing — 2004. — pp. 404-411.

27. Mikolov T. *Efficient Estimation of Word Representations in Vector Space* / T. Mikolov, K. Chen, G. Corrado, J. Dean // arXiv.org. — 2013. [Электронный ресурс] — URL: https://arxiv.org/abs/1301.3781.

28. Miller D. *Leveraging BERT for Extractive Text Summarization on Lectures*. / D. Miller // arXiv.org. — 2019. [Электронный ресурс] — URL: https://arxiv.org/abs/1906.04165.

29. Nenkova A. *Automatic summarization* / A. Nenkova, K. McKeown // Foundations and Trends in Information Retrieval Vol. 5. — 2011. — pp. 103-233.

30. Pennington J. GloVe: *Global Vectors for Word Representation* / J. Pennington, R. Socher, C. D. Manning // Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing ({EMNLP}) — 2014. — pp. 1532-1543.

31. Peters M. E. *Deep contextualized word representations* / M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer // arXiv.org. — 2018. [Электронный ресурс] — URL: https://arxiv.org/abs/1802.05365.

32. Qaiser S. *Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents* / S. Qaiser, R. Ali // International Journal of Computer Applications — 2018. [Электронный ресурс] — URL: https://www.researchgate.net/publication/326425709_Text_Mining_Use_of_TF-IDF_to_Examine_the_Relevance_of_Words_to_Documents.

33. R. Garc´ıa-Hern´andez, R. Montiel, Y. Ledeneva, E. Rend´on, A. Gelbukh and R. Cruz. *Text Summarization by Sentence Extraction Using Unsupervised Learning* // In Proceedings of the 7th Mexican International Conference on Artificial Intelligence: Advances in Artificial Intelligence/ Alexander Gelbukh and Eduardo F. Morales (Eds.) — Springer-Verlag, Berlin, Heidelberg, 2008. — pp. 133-143.

34. Radford A. *Language Models are Unsupervised Multitask Learners* / A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever // OpenAI Blog — 2019. [Электронный ресурс] — URL: https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf .

35. Ramos J. *Using TF-IDF to Determine Word Relevance in Document Queries* / J. Ramos // Proceedings of the first instructional conference on machine learning — 2003.

36. Reimers N. *Making Monolingual Sentence Embeddings Multilingual using Knowledge Distillation*. / N. Reimers, I. Gurevych // arXiv.org. — 2020. [Электронный ресурс] — URL: https://arxiv.org/abs/2004.09813.

37. Reimers N. *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks*. / N. Reimers, I. Gurevych // arXiv.org. — 2019. [Электронный ресурс] — URL: https://arxiv.org/abs/1908.10084.

38. Repository of pretrained models [Электронный ресурс] — URL: http://docs.deeppavlov.ai/en/master/intro/pretrained_vectors.html.

39. See A. *Get To The Point: Summarization with Pointer-Generator Networks* / A. See, P. J. Liu, C. D. Manning // arXiv.org. — 2017. [Электронный ресурс] — URL: https://arxiv.org/abs/1704.04368.

40. Semantic Textual Similarity Benchmark [Электронный ресурс] — URL: http://ixa2.si.ehu.es/stswiki/index.php/Main_Page.

41. Sennrich R. *Neural Machine Translation of Rare Words with Subword Units* / R. Sennrich, B. Haddow, A. Birch // arXiv.org. — 2016. [Электронный ресурс] — URL: https://arxiv.org/abs/1508.07909.

42. Sharma E. *BIGPATENT: A Large-Scale Dataset for Abstractive and Coherent Summarization* / E. Sharma, C. Li, L. Wang // arXiv.org. — 2019. [Электронный ресурс] — URL: https://arxiv.org/abs/1906.03741.

43. Subramanian S. *Extractive and Abstractive Neural Document Summarization with Transformer Language Models*. / S. Subramanian, R. Li, J. Pilault, C. Pal // arXiv.org. — 2020. [Электронный ресурс] — URL: https://arxiv.org/abs/1909.03186.

44. Vaswani A. *Attention Is All You Need* / A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin // arXiv.org. — 2017. [Электронный ресурс] — URL: https://arxiv.org/abs/1706.03762.

45. Wolf T. *The Big & Extending Repository of pretrained Transformers*. // 2019. [Электронный ресурс] — URL: https://github.com/huggingface.

Приложение А. Оценки качества работы алгоритмов

Алгоритм\Оценка	ROUGE-1	ROUGE-2	ROUGE-L	Key words
CSBERT_KMeans	0.2732	0.0724	0.1749	0.6438
CSBERT_PRank	0.1709	0.0267	0.1218	0.4815
FT_KMeans	0.2747	0.0714	0.1783	0.6344
FT_PRank	0.1004	0.0202	0.0848	0.4656
RuBERT_KMeans	0.2793	0.0742	0.1815	0.6400
RuBERT_PRank	0.1422	0.0471	0.1169	0.5227
RuSBERT_KMeans	0.2632	0.0674	0.1715	0.6278
RuSBERT_PRank	0.1932	0.0691	0.1526	0.5900
Simple_PRank	*0.3075*	*0.0833*	*0.1885*	*0.6947*
TextRank	**0.3436**	**0.1153**	**0.2157**	**0.7159**
TF-IDF_KMeans	*0.2936*	*0.0816*	*0.1878*	*0.6890*
Mixed	**0.3383**	**0.1098**	**0.2079**	**0.7194**

Таблица 1. Качество работы алгоритмов, автоматическое оценивание

Алгоритм\Оценка	Humanity	Science	Tech	Общая оценка
CSBERT_KMeans	*38*	*36.5*	*27*	*101.5*
CSBERT_PRank	11	18.5	13	42.5
FT_KMeans-Clean	30.5	25	16	71.5
FT_KMeans-Raw	26.5	21	15	62.5
FT_PRank-Clean	3	9.5	1	13.5
FT_Prank-Raw	3	10	1	14
RuBERT_KMeans	24	31.5	20	75.5
RuBERT_KMeans-ST	29.5	33.5	16.5	79.5
RuBERT_PRank	4	6.5	2	12.5
RuBERT_Prank-ST	5.5	4	5	14.5
RuSBERT_KMeans	25.5	31	14.5	71
RuSBERT_KMeans-ST	25.5	33.5	23	82
RuSBERT_PRank	2.5	7.5	1	11
RuSBERT_Prank-ST	2.5	5.5	1.5	9.5
Simple_PRank	*36.5*	*37.5*	*33*	*107*
TextRank	*36.5*	*35.5*	*35.5*	*107.5*
TF-IDF_KMeans	29	29.5	23	81.5

Mixed	**41**	**43.5**	**37**	**121.5**

Таблица 2. Качество работы алгоритмов, экспертное оценивание (с удалением коротких предложений)

Алгоритм\Оценка	Humanity	Science	Tech	Общая оценка
CSBERT_KMeans	30	23	18	71
CSBERT_PRank	18	6	6.5	30.5
FT_KMeans-Clean	22	16	6.5	44.5
FT_KMeans-Raw	19.5	23.5	6.5	49.5
FT_PRank-Clean	14.5	6	0	20.5
FT_Prank-Raw	15	6.5	1	22.5
RuBERT_KMeans	26.5	25.5	15.5	67.5
RuBERT_KMeans-ST	23.5	31	7.5	62
RuBERT_PRank	15	11	7.5	33.5
RuBERT_Prank-ST	15	25	7	47
RuSBERT_KMeans	18	22.5	11	51.5
RuSBERT_KMeans-ST	24.5	16	12	52.5
RuSBERT_PRank	22	20.5	13.5	56
RuSBERT_Prank-ST	22	25	14.5	61.5
Simple_PRank	*33*	*30.5*	*26*	*89.5*
TextRank	**31.5**	**29.5**	**30.5**	**91.5**
TF-IDF_KMeans	*32*	*32*	*17.5*	*81.5*

Таблица 3. Качество работы алгоритмов, экспертное оценивание (без удаления коротких предложений)

Приложение В. Примеры работы модели «Mixed»

Пример 1 («гуманитарный» текст). Вопросы выявления и сохранения национальной идентичности в архитектуре Сибири (на примере Республики Хакасия).

Эталонный реферат

Предметом исследования настоящей статьи является проявление национальной политики в архитектуре Сибири. Объект исследования – архитектура городов Республики Хакасия (на примере города Абакана). Хронологические границы исследования - с момента установления советской власти до 1950-х годов. Цель исследования – на основе изучения национальной политики, проводимой властями в регионах страны советского периода, выявить объекты архитектуры с проявлением национальной специфики, представляющие собой ценность и являющиеся перспективными для сохранения как объекты культурного наследия. В статье подробно рассматриваются такие аспекты темы как проявление политики "коренизации" во всех сферах культурно-бытовой жизни советского общества, развитие национального самовыражения в архитектуре страны в соответствии с национальной политикой, проводимой властями. В исследовании использованы историографический анализ и натурное обследование застройки городов одной из национальных автономий России. Применены историко-эволюционный и сравнительно-типологический метод при изучении национальной политики, проводимой в СССР, и взаимодействия национального своеобразия с архитектурой в контексте истории архитектуры страны, а также методики определения историко-культурной ценности объектов. Научная новизна исследования заключается в выявлении методов и средств отражения национальной специфики объектов архитектуры национальных территорий СССР. Установлено, что политика "коренизации" выражалась в архитектуре созданием объектов с национальной спецификой методами "стилизации" и "формообразования". Выявлен наиболее интересный архитектурный объект – жилой дом специалистов в городе Абакане, обладающий национальными чертами, являющийся уникальным и ценным для рассматриваемой территории (Республики Хакасия). Результаты исследования могут быть использованы при изучении архитектуры национальных территорий.

Сгенерированный реферат

Целью исследования является выявление объектов архитектуры, которые имеют проявление национальной специфики, представляющие собой ценность и перспективны для сохранения. Объектом настоящего исследования является архитектура одной из национальных автономий России – Республики Хакасия. В работе использованы историко-эволюционный и сравнительно-типологический метод при изучении национальной политики, проводимой в СССР и взаимодействия национального своеобразия с архитектурой в контексте истории архитектуры страны. Течения архитектуры авангарда, отвергавшие преемственность архитектуры прошлого,

одновременно существовали с национально стилизованной архитектурой. а – проект жилых домов для оседающих кочевников в Киргизской республике, макет, архитектор В. Калмыков, 1933-1934 гг.; б – проект многоэтажного жилого дома для оседающих кочевников в Киргизской республике, макет, архитектор В. Калмыков, 1933-1934 гг.; в – проект жилых домов для оседающих кочевников в Киргизской республике (фасады, планы, разрезы, аксонометрия), архитектор В. Калмыков, 1933-1934 гг.; г – проект жилища для городов Средней Азии, жилой дом – тип № 2 (планы первого и второго этажей, разрез жилой ячейки), архитекторы В. Лавров, В. Попов, 1931 г.; г – проект блокированных домов для безлесных районов Казахстана (аксонометрия), А. Бунин, М. Круглова, 1929-1930 гг.; д – проект блокированных домов для безлесных районов Казахстана (план, разрез, торцевой фасад), А. Бунин, М. Круглова, 1929-1930 гг. Таким образом, проявление национальной специфики в архитектуре в 1920–1930-х гг. условно можно разделить на "стилизацию" с использованием традиционных форм и национального декоративного искусства (орнамента) в рамках основного архитектурного стиля и "формообразование" новой национальной архитектуры, ориентированной на климатические, природно-географические, культурные и социально-бытовые факторы региона. При натурном исследовании и изучении проявления национальной специфики в архитектуре городов Республики Хакасия, был выявлен объект – жилой дом специалистов, построенный в Абакане. В целом национальная политика проводилась во всех национальных субъектах страны, затрагивая различные сферы жизни общества. В ходе исследования городов Республики Хакасия выявлен один из ярких эклектичных примеров с элементами национальной темы – жилой дом специалистов в городе Абакане.

Пример 2 («технический» текст). Разработка плагина «Портфолио СибГИУ» для системы управления обучением «Moodle»

Эталонный реферат

Объектом исследования является электронное портфолио студента, а предметом исследования – разработка информационной системы, позволяющей формировать электронное портфолио, характеризующее индивидуальные достижения обучающегося по шести направлениям деятельности: "Учебная деятельность по основной образовательной программе"; "Научно-исследовательская деятельность"; "Прочие образовательные достижения"; "Общественная деятельность"; "Культурно-творческая деятельность"; "Спортивная деятельность". Для эффективной работы информационной системы требовалось организовать разграничение прав доступа для различных групп пользователей: студентов, модераторов портфолио и администраторов. Информационная система реализована в виде отдельного модуля (плагина) для системы управления обучением "Moodle". При создании информационной системы "Портфолио СибГИУ" использовался метод дедукции, при котором по множеству частных признаков делается заключение об общей совокупности исследуемых признаков, а также метод анализа существующего плагина Exabis E-Portfolio. Основным результатом работы является разработка и программная реализация информационной системы "Портфолио СибГИУ", которая в настоящее время внедрена

и является частью электронной информационно-образовательной среды Сибирского государственного индустриального университета. Важная особенность системы заключается в динамическом формировании в портфолио категории "Учебная деятельность по основной образовательной программе" путем синхронизации выложенных в Moodle работ (файлов) студента, результатов прохождения тестов в Moodle, а также отзывов и итоговых оценок на выполненные задания, выставленные преподавателем в электронном курсе Moodle. Использование информационной системы показало, что работа в системе не представляет трудностей ни для студентов, ни для модераторов. Обучающийся имеет возможность загрузить в портфолио документы, подтверждающие личные достижения в различных областях, при этом каждое его действие контролируется модератором, что позволяет повысить качество наполнения и избежать ошибок при формировании портфолио. Информационная система "Портфолио СибГИУ" удовлетворяет требованиям ФГОС 3+ и обеспечивает накопление, систематизацию и учет комплекта электронных документов, характеризующих индивидуальные достижения обучающегося по различным направлениям деятельности.

Сгенерированный реферат

В результате в университете было принято решение реализовать электронное портфолио в виде информационной системы "Портфолио СибГИУ", разработанной в качестве дополнительного модуля (плагина) типа блок к системе Moodle [11]. Информационная система "Портфолио СибГИУ" должна обеспечивать накопление, систематизацию и учет комплекта электронных документов, характеризующих индивидуальные достижения обучающегося по различным направлениям деятельности и реализовывать следующий функционал:. При формировании шапки и подвала на веб-страницах в плагине портфолио применяется Output API [9]. Данное API позволяет через глобальную переменную $PAGE сформировать навигационную панель Moodle ("Хлебные крошки"), установить заголовки страницы, подключать к ней стили и JavaScript. Рисунок 1 - Блок доступа к электронному портфолио на главной странице авторизированного пользователя в системе Moodle. Раздел "Учебная деятельность по ООП" в электронном портфолио формируется динамически, путем импорта всех работ (файлов), загруженных студентами в электронный курс в системе Moodle, а также оценок, выставленных за выполнение этих работ и отзывов (рецензий) преподавателей на эти работы. Далее из сформированных таким образом данных необходимо исключить те курсы, в которых обучающийся не сдал ни одного задания, а также те задания и тесты, которые не были выполнены. Обучающийся имеет возможность загрузить в портфолио документы, подтверждающие личные достижения в различных областях, при этом каждое его действие контролируется модератором, что позволяет повысить качество наполнения и избежать ошибок при формировании портфолио.

Пример 3 («научный/социально-экономический» текст). Оценка внедрения публичного управления в муниципальную практику (на примере городов и муниципальных районов Волгоградской области).

Эталонный реферат

Объектом исследования является совокупность факторов, конституирующих систему публичного управления в регионах Российской Федерации, реализация которой повышает результативность государственного и муниципального управления за счет роста прозрачности процесса принятия управленческих решений и осуществления общественного контроля за их реализацией. Автором предложена модель реализации публичного управления, отражающая процесс взаимодействия его субъектов, а также на примере городов и муниципальных регионов Волгоградской области дана количественная характеристика ресурсных, инфраструктурных, институциональных и результирующих факторов, определяющих степень внедрения публичного управления в муниципальную практику. В рамках фундаментального системно-функционального подхода к исследуемому объекту были применены следующие аналитические методы: структурный, комплексный, сравнительный и монографический. Выводы работы основываются на результатах использования таких частных методов анализа, как структурирование, сопоставление и ранжирование. В статье представлено теоретическое обоснование сущности публичного управления и определены факторы, конституирующие его реализацию. В результате проведенного исследования сформирована совокупность индикаторов, характеризующих ресурсные, инфраструктурные и институциональные факторы реализации системы публичного управления, а также определены показатели, оказывающие наибольшее влияние на внедрение технологий публичного управления в муниципальную практику Волгоградской области.

Сгенерированный реферат

Усугубляется проблема износа основных фондов сокращением долей расходов на инвестиции в основной капитал и на ввод в действие основных фондов в ВВП России, которые за 1990-е гг. снизились более чем в два раза (таб. - число участников в реализации государственной закупочной политики (количество поданных на конкурс заявлений) − позволяет оценить готовность бизнес-структур участвовать в закупочной деятельности государства на предлагаемых условиях, то есть отражает участие организаций бизнеса в реализации управленческих решений, направленных на использование общественных ресурсов, и, следовательно, может быть использовано для характеристики результирующих факторов публичного управления. - степень использования Интернета организациями – характеризует использование интернет-коммуникаций организациями;. В первую очередь рассмотрим обеспеченность МР и ГО ресурсами для внедрения технологий публичного управления (таб. Для успешной реализации технологий публичного управления в муниципальной практике необходимо наличие не только социально-экономических и технологических условий, но и должна быть сформирована соответствующая институциональная среда для участия и контроля граждан над решением вопросов, затрагивающих их интересы, оценить которую возможно на основании количественного анализа показателей институциональных факторов (таб. Количественный анализ, построенный на отборе МО и ГО, в которых значение показателей результирующих факторов максимально, позволил выявить муниципальные образования Волгоградской области, для которых характерна более высокая

степень внедрения технологий публичного управления в муниципальную практику:. Таким образом, предложенная автором модель оценки степени публичного управления позволяет осуществлять мониторинг его реализации в муниципальных образованиях, результаты которого могут быть использованы для разработки целевых мероприятий по внедрению публичного управления в Российской Федерации, что обеспечит рост эффективности использования общественных ресурсов за счет повышения прозрачности процессов принятия и реализации решений в системе государственного и муниципального управления и, следовательно, обеспечит возможности для перехода к инновационной модели развития.

Пример 4 (литературное произведение с последовательным развитием событий). Р. Брэдбери «И грянул гром».

Сгенерированный реферат

Объявление на стене расплылось, словно его затянуло пленкой скользящей теплой воды; Экельс почувствовал, как веки, смыкаясь, на долю секунды прикрыли зрачки, но и в мгновенном мраке горели буквы: А/О САФАРИ ВО ВРЕМЕНИ ОРГАНИЗУЕМ САФАРИ В ЛЮБОЙ ГОД ПРОШЛОГО ВЫ ВЫБИРАЕТЕ ДОБЫЧУ МЫ ДОСТАВЛЯЕМ ВАС НА МЕСТО ВЫ УБИВАЕТЕ ЕЕ В глотке Экельса скопилась теплая слизь; он судорожно глотнул. Он скажет вам, где и когда стрелять. Слава богу, что победил Кейт. - Вот именно, - отозвался человек за конторкой. Мы организуем сафари. Экельс вспыхнул от возмущения. В машине было еще четверо. Тревис - руководитель сафари, его помощник Лесперанс и два охотника - Биллингс и Кремер. Таких мы не трогаем. - Господи, - произнес Экельс. Машина остановилась. - Почему? - спросил Экельс. - Я что-то не понимаю, - сказал Экельс. - Не будет потомков от потомков от всех ее потомков! - Хорошо, они сдохли, - согласился Экельс. Десятью лисами меньше - подохнет от голода лев. Наступите на мышь - и вы оставите на Вечности вмятину величиной с Великий Каньон. Никогда не сходите с нее! - Понимаю, - сказал Экельс. Малейшее отклонение сейчас неизмеримо возрастет за шестьдесят миллионов лет. Мы не знаем - только гадаем. Тревис и Лесперанс переглянулись. Но мы не видели ничего. Экельс бледно улыбнулся. - Ну, все, - отрезал Тревис. - Даже в шутку не цельтесь, черт бы вас побрал! И не сходите с Тропы. - Странно, - пробормотал Экельс. - Приготовиться! - скомандовал Тревис. - Я дрожу, как мальчишка. - Тихо, - сказал Тревис. Безбрежные джунгли были полны щебета, шороха, бормотанья, вздохов. Оно за тридцать футов возвышалось над лесом - великий бог зла, прижавший хрупкие руки часовщика к маслянистой груди рептилии. Оно сомкнуло челюсти в зловещем оскале. - Губы Экельса дрожали. - Он еще не заметил нас. - Кру-гом! - скомандовал Тревис. На сей раз я просчитался. Это мне не по силам. Спрячьтесь в Машине. - Казалось, Экельс окаменел. Он застонал от бессилия. Ружья дали еще залп. Чудовище лежало неподвижно. Колыхаясь, оно приняло покойное положение. С гулом он обрушился на безжизненное чудовище, как бы окончательно утверждая его гибель. Но мы можем сфотографировать вас возле нее. - Простите меня, - сказал он. - Вы не вернетесь с Машиной. Вы останетесь здесь! Лесперанс перехватил руку Тревиса. - Тревис стряхнул его руку. Но главное даже не это. Он соскочил с Тропы.

Понимаешь, чем это нам грозит? Мы гарантируем, что никто не сойдет с Тропы. - Мы ничего не знаем! Экельс полез в карман. - Это несправедливо! Больше его не потянет за такой добычей. - Не глядите на меня, - вырвалось у Экельса. - Я ничего не сделал. Чего вы от меня хотите? 2055. Машина остановилась. - Выходите, - скомандовал Тревис. Хотя нет, не совсем такая же. - О'кей, Экельс, выходите. Экельс будто окаменел. - Ну? - поторопил его Тревис. - Что вы там такое увидели? А тут еще это ощущение. - Нет, не может быть! Из-за такой малости. Мысли Экельса смещались. Он лежал неподвижно.