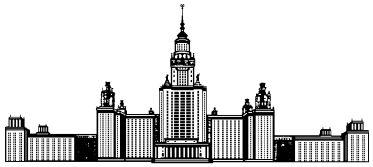
Московский государственный университет имени М. В. Ломоносова



Факультет Вычислительной Математики и Кибернетики

Кафедра Математических методов прогнозирования

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

«Нейросетевые модели языка для ранжирования фраз в полуавтоматической суммаризации научных статей»

Выполнила:

студентка 4 курса 417 группы

Дзюба Мария Эдуардовна

Научный руководитель:

д.ф-м.н., профессор РАН

Воронцов Константин Вячеславович

Москва, 2023

Содержание

1 Введение

С каждым годом объем научных исследований многократно увеличивается, что приводит к значительному повышению нагрузки на ученных в момент проведения ими предварительного исследования и изучения предложенных раннее методов решения поставленных задач.

При написании научной работы важно структурировать информацию, а также размещать исследованную литературу в порядке, который будет наиболее понятен читателю.

В данном случае можно опираться на ранее написанные статьи по определенной тематике для выяления основных тенденций и приемов, которые используются для ранжирования связанных статей.

Системы суммаризации особенно актуальны для учёных и экспертов, которые вынуждены тратить большое количество времени на чтение научных публикаций. В настоящей работе рассматривается задача полуавтоматической суммаризации, цель которой заключается в оказании пользователям помощи при написании авторского обзора (реферата, дайджеста) по заданной тематической подборке.

Автоматическое реферирование или суммаризация – создание краткой версии исходного текста с описанием его ключевых идей.

Актуальность данной работы заключается в исследовании методов ранжирования связанных статей при написании собственного исследования. При этом данные методы могут быть использованы на одном из первых этапов полуавтоматической суммаризации текстов.

Практическая значимость данной работы заключается в том, что реализация предложенной системы может быть встроена в поисково-рекомендательную систему формирования и анализа тематических подборок англоязычных научных статей «Мастерская знаний».

В текущей реализации поисково-рекомендательной системы пользователям предлагается самостоятельно выстроить порядок цитируемых документов. Добавление одного или нескольких из предложенных алгоритмов поможет исследователям выбирать наиболее подходящий под их логику повествования метод ранжирования и

логично располагать уже изученные статьи в ходе написания собственного исследования.

2 Постановка задачи

Пусть D - коллекция научных публикаций, содержащих ссылки на публикации из множества C. Для каждого документа $d \in D$ выделена последовательность из k_d ссылок $X_d = (x_{d1}, ..., x_{dk})$, $x_{di} \in C$. Задача состоит в том, чтобы по этим данным построить модель ранжирования ссылок, которая принимает на вход произвольное неупорядоченное подмножество $X \subset C$ и выдает на выходе отношение линейного порядка на X, наиболее согласованное с наблюдаемыми последовательностями $X_D = \{X_d : d \in D\}$.

Согласованность модели ранжирования $f(x)$ с последовательностью X_d предлагается измерять с помощью коэффициента корреляции Кендалла, равного доли правильно ранжированных пар:

$$\tau_f(X_d) = \frac{2}{k_d(k_d - 1)} \sum_{i<j} [f(x_{di}) < f(x_{dj})].$$

Согласованность модели ранжирования $f(x)$ со всей совокупностью наблюдаемых данных X_D определяется путем усреднения $\tau_f(X_d)$ по всем последовательностям X_d:

$$\tau_f(X_D) = \frac{1}{|D|} \sum_{d \in D} \tau_f(X_d).$$

Коэффициент ранговой корреляции Кендалла принимает значения в отрезке [0 , 1], чем выше значение, тем лучше.

Для построения параметрических моделей ранжирования $f(x, w)$ удобно минимизировать число неверно отранжированных пар (i, j). При этом, будем стремиться минимизировать также и количество пар с одинаковым выходом функции, тем самым повышая однозначность итоговой расстановки.

$$Q(x, w) = \sum_{d \in D} \sum_{i<j} [f(x_{dj}, w) - f(x_{di}, w) \leq 0] \to \min_w$$

Функция $M_{ij} = f(x_{dj}, w) - f(x_{di}, w)$ называется *парным отступом* и используется для определения аппроксимированных оптимизационных критериев в попарных

методах обучения ранжированию:

$$Q(x, w) \leq \tilde{Q}(x, w) = \sum_{d \in D} \sum_{i<j} \mathcal{L}(M_{ij}(w) \to \min_{w},$$

где $\mathcal{L}(M) \geq [M \leq 0]$ - гладкая верхняя оценка функции потерь.

2.1 Определения и обозначения

Граф цитирований - граф, каждая вершина которого представляет документ в коллекции научных статей; каждое ребро направлено от одного документа к другому, который он цитирует.

Модель ранжирования - функция $f : C \to \mathbb{R}$, с помощью которой элементы заданного неупорядоченного подмножества ссылок $X = \{x_1, ..., x_k\} \subset C$ могут быть отранжированы по возрастанию $f(x^{(1)}) \leq f(x^{(2)}) \leq ... \leq f(x^{(k)})$.

2.2 Обзор литературы

Существует множество методов для ранжирования ссылок, которые можно разделить на две категории: методы на основе графов и методы на основе машинного обучения.

2.2.1 Методы на основе графов

В поставленной задаче можем рассматривать цитирования как ссылки. Учитывая факт общедоступности графа цитирований, можем применить описанные ниже методы.

Методы на основе графов используют структуру ссылок между документами для определения их важности. Одним из таких методов является алгоритм PageRank[5], который был разработан для ранжирования веб-страниц. Он определяет важность страницы на основе количества ссылок, указывающих на нее, и значимость страниц, ссылающихся на нее. Аналогичные алгоритмы могут быть применены для ранжирования ссылок в задаче суммаризации.

Также были предложены различные подходы Weighted PageRank [6], [7], [8] - модификация алгоритма PageRank, которая учитывает не только количество ссылок,

указывающих на страницу, но и их вес. Вес ссылки может зависеть от различных факторов, таких как авторитетность сайта, на котором размещена ссылка, релевантность текста ссылки и т.д.

В уже существующих исследованиях были предложены следующие факторы, относительно которых выставляется вес для цитаты: Yu, Li и Liu (2004) [4] предложили добавлять вес каждой вершине графа в зависимости от года публикации работы.

2.2.2 Методы на основе машинного обучения

Одним из основных медотов ранжирования с помощью машинного обучения является использование градиентного бустинга над решающими деревьями. Наиболее популярными имплементациями являются XGBoost [2] и LightGBM [1]. Для дальнейшей работы был выбран алгоритм LightGBM из-за наличия возможности использования персонализированной оптимизационной функции.

3 Описание данных и признакового пространства

3.1 Используемые данные

В качестве данных предлагается использовать данные из коллекции S2ORC (The Semantic Scholar Open Research Corpus) [3]. В корпусе представлены данные о 81.1M статьях из разных научных областей, при этом полные тексты документов представленны для 8M статей. В данном наборе данных объединены статьи из сотен академических издательств и цифровых архивов. На сегодняшний момент эта коллекция является самой большой общедоступной коллекцией машиночитаемых академических текстов. Статьи данного корпуса объединены в граф цитирования, содержащий 467M вершин.

Особенности данной коллекции документов:

1. В коллекции предоставляются метаданные о статьях, такие как авторы, аннотации, ключевые слова и ссылки на другие исследования.

2. Для каждой статьи представлена аннотация.

3. Текст статьи разделен на абзацы, внутри которых выделены цитирования, приложенные таблицы и рисунки.

4. Цитаты внутри абзацей связаны с элементами библиографии, приведенной в метаданных статьи, что позволяет однозначно сопоставить цитируемую статью с ее вершиной в графе цитирований.

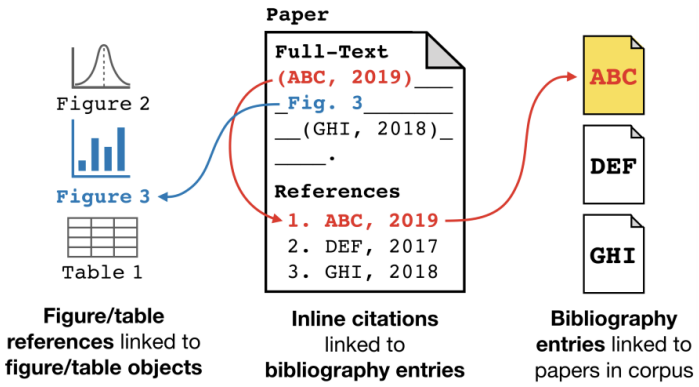


Рис. 1: Общий вид данных, представленных в выбранном корпусе

Общий объем данной коллекции составляет более 1Тб данных, поэтому в данной работе используется только его часть - ACL (Anthology of Computer Linguistics)[9]. Все статьи настоящего раздела посвящены компьютерным наукам и обработке естественного языка. Объем данной части S2ORC составляет 41к статей. После предварительной обработки - удаления повторяющихся статей, удаления статей без полного текста работы, а также удаления статей без полных метаданных - для дальнейшего использования доступны 14к статей. При этом, вследствие неполноты данных о некоторых цитируемых статьях также предлагается удалить из выборки подмножество исходных статей с количеством цитируемых статей меньше трёх.

3.2 Построение выборки

В качестве объекта выборки рассматривается пара <Исходная статья, Связанная статья>, а также характеристики каждой из статей.

Для сбора данных в необходимом для дальнейшего обучения моделей машинного обучения виде предлагается следующий алгоритм - среди всех представленных

абзацей статьи выделяем все, связанные с введением и обзором литературы, и объединяем их в соответсвующие части; внутри каждой части выделяем цитирования и однозначно сопоставляем их с элементами библиографии.

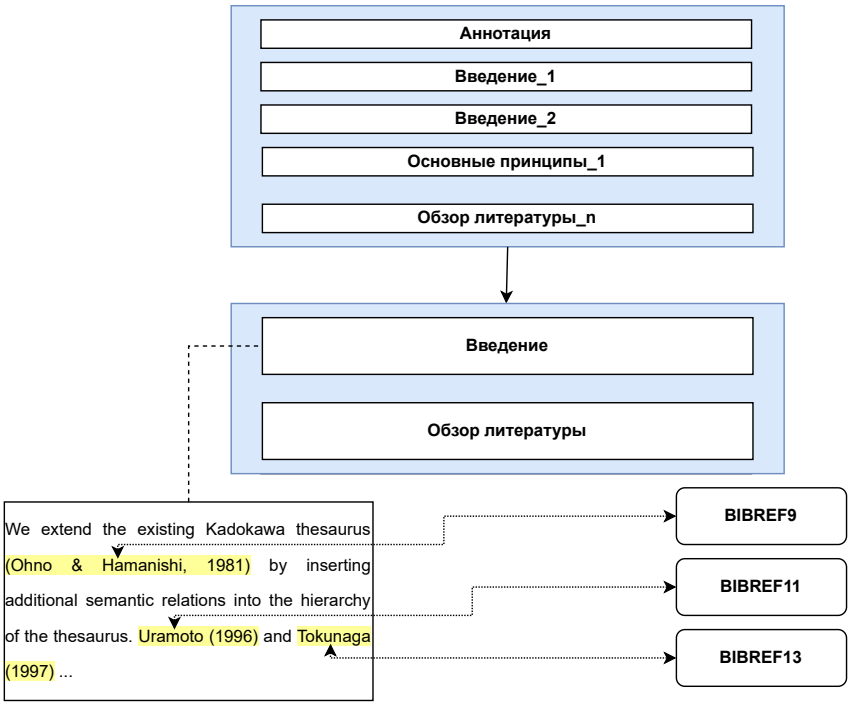


Рис. 2: Алгоритм построения выборки

После сбора данных вся выборка разбивается на тренировочную и тестовую в пропорциях 90:10. При этом в обучающей выборке содержаться 12075 исходных статей, а в тестовой - 1118 статей.

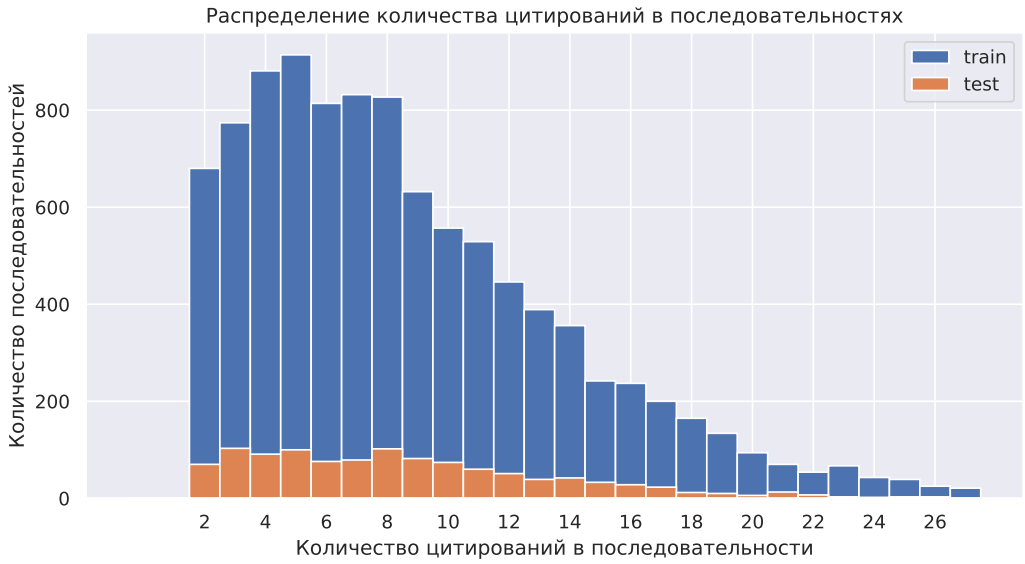


Рис. 3: Распределение количества цитирований в последовательности

Как мы можем видеть из графика - тестовая выборка является репрезентативной: в ней отражены последовательности ссылок всех представленных в исходном наборе данных длин.

3.3 Признаковое пространство

Для дальнейшего обучения моделей машинного обучения предлагается собрать данные, которые будут удовлетворять двум условиям - наиболее полно описывать связанные документы и их взаимосвязь с исходной статьей; быть доступными на этапе применения методов в реальной жизни. Предлагается использовать следующий набор признаков:

- год публикации - есть в метаданных
- количество цитирований - есть в метаданных
- количество важных цитирований - есть в метаданных
- минимальное количество публикаций у авторов статьи - из графа цитирований
- минимальное количество цитирований у авторов статьи - из графа цитирований

- максимальное соотношение количества публикаций с количеством цитирований у авторов статьи - из графа цитирований
- максимальное количество публикаций среди авторов статьи - из графа цитирований
- максимальное соотношение количества публикаций с количеством цитирований у авторов статьи - из графа цитирований

Данные признаки подходят под описанные выше требования, так как всю необходимую для их построения информацию можно получить из графа цитирований и метаданных о статье.

4 Модели для решения поставленной задачи

4.1 Базовая модель

В качестве базовой модели рассматривается ранжирование связанных документов по году публикации от наиболее старых к наиболее новым статьям. При этом, в случае неоднозначности, считается, что документы с одним годом публикации получают один и тот же выход алгоритма.

4.2 Модели ранжирования на основе алгоритма PageRank

Отличительной характеристикой использования PageRank является возможность использования корпуса статей во много раз превышающего объемы, доступные для работы при использовании других алгоритмов. Это связано с тем, что все вычисления проводятся только с использованием информации из графа цитирований. При этом в данном случае уходит необходимость наличия полного или частично доступного текста, что сильно расширяет объем доступной выборки.

При этом в некоторых статьях цитирование одной работы происходит в нескольких местах. В данном случае не будем использовать информацию о многочисленных вхождениях и будем учитывать только единственное вхождение.

4.2.1 PageRank

Алгоритм PageRank может быть представлен в виде следующего уравнения:

$$PR(x_k) = \frac{(1 - coeff)}{D} + coeff \cdot \sum_{i=1}^{n} \frac{PR(x_i)}{C_i},$$

где n - количество ссылающихся на документ x статей, $coeff$ - коэффицент затухания (положим его равным 0.85), D - общее число документов, C_i - количество ссылок, ссылающихся на i-й документ.

Для оценки PageRank для представленных статей будем использовать граф цитирований.

4.2.2 Weighted Pagerank

Алгоритм Weighted PageRank может быть представлен в виде следующего уравнения:

$$PR_w(x_k) = (1 - coeff) \cdot \frac{w(x)}{\sum_{n=1}^{N} w(x_n)} + coeff \cdot \sum_{i=1}^{n} \frac{PR_w(x_i)}{w_i},$$

где n - количество ссылающихся на документ x статей, $coeff$ - коэффицент затухания (положим его равным 0.85), w_i - вес i-й публикации.

Для оценки Weighted PageRank для представленных статей будем использовать граф цитирований. При этом, в качестве веса попробуем использовать следующие статистики, которые можно получить из метаданных о статье:

1. Использование года публикации в качестве веса статьи;
2. Использование количества цитирований конкретного объекта в качестве веса статьи.

4.2.3 Общая имплементация

Алгоритм PageRank (аналогично weighted Pargerank) работает следующим образом:

1. Создается матрица связей между страницами, где каждый элемент матрицы указывает на наличие ссылки между двумя страницами.

2. Создается вектор начальных значений, где каждый элемент равен 1/D, где D - общее количество страниц.

3. Вычисляется матрица переходов, которая представляет собой взвешенную матрицу связей, где каждый элемент равен 1/количество ссылок на данной странице.

4. Производится итерационный процесс, в котором вектор значений умножается на матрицу переходов. Этот процесс повторяется до тех пор, пока значения вектора не стабилизируются.

5. Полученный вектор значений является ранжированием страниц, где страницы с более высоким значением более важны и авторитетны.

4.3 Модели ранжирования на основе статистики совместного распределения

Рассмотрим последовательность ссылок $(x_1, ..., x_k)$. Назовем дистанцией между эллементами x_i и x_j в этой последовательности разность $\delta_{x_i x_j} = j - i$. Дистанция δ_{uv} положительна, если u располагается левее v и отрицательна, если правее.

Обозначим через $R_{uv}(D)$ мультимножество значений дистанций δ_{uv}, наблюдаемых во всех последовательностях X_d, $d \in D$ для данной пары $(u, v) \in C^2$.

Ссылки в последовательности могут повторяться. По умолчанию будем полагать, что в случае повторных ссылок в $R_{uv}(D)$ заносятся только первые вхождения ссылок в последовательность.

По мультимножеству $R_{uv}(D)$ определяются эмперические оценки функции распределения, математического ожидания и дисперсии дистанций δ_{uv}:

$$F_{uv}(z) = \frac{1}{|R_{uv}|} \sum_{\delta \in R_{uv}} [\delta \leq z] - \text{выборочная функция распределения;}$$

$$\mu_{uv} = \frac{1}{|R_{uv}|} \sum_{\delta \in R_{uv}} \delta - \text{оценка средней дистанции;}$$

$$\sigma^2_{uv} = \frac{1}{|R_{uv}|} \sum_{\delta \in R_{uv}} (\delta - \mu_{uv})^2 - \text{оценка среднеквадратичного отклонения.}$$

4.3.1 Простейшие эвристические модели ранжирования

Базовая эвристическая модель ранжирования $f_0(x)$ основана на том, чтобы посчитать по всей выборке, насколько чаще элемент x находится правее всех остальных элементов заданного (неупорядоченного) множества X, чем левее:

$$f_0(x) = \sum_{u \in X \setminus x} \frac{1}{|R_{uv}|} \sum_{\delta \in R_{uv}} [\delta \geq 0].$$

Второй вариант - учесть дистанцию, насколько x правее других элементов последовательности:

$$f_\delta(x) = \sum_{u \in X \setminus x} \frac{1}{|R_{uv}|} \sum_{\delta \in R_{uv}} \delta[\delta \geq 0].$$

Третий вариант - учесть все дистанции между x и другими объектами вне зависимости от их взаимного расположения. При этом, если x стоит правее объекта, то дистанция между ними добавляется с положительным знаком, если левее - с отрицательным:

$$f_{\delta_{all}}(x) = \sum_{u \in X \setminus x} \frac{1}{|R_{uv}|} \sum_{\delta \in R_{uv}} \delta.$$

Четвертый вариант - учесть среднюю дистанцию между x и другими элементами последовательности:

$$f_\mu(x) = \sum_{u \in X \setminus x} \mu_{ux}.$$

Пятый вариант - попытка учесть, что большое значение средней дистанции μ_{ux} может быть связанно с большим разбросом σ_{ux} случайно, а не потому, что x значимо чаще оказывается правее u. Предлагается ввести нормировку на среднеквадратичное отклонение:

$$f_\sigma(x) = \sum_{u \in X \setminus x} \frac{\mu_{ux}}{\sigma_{ux}}$$

Стоит отметить, что перечисленные выше модели обладают затрудняющим работу свойством - их выход сильно зависит от тех документов, которые мы использовали в обучении. По сути у них нет обобщающей способности на новые документы, поэтому для неизвестных документов выход всех моделей одинаков и равен 0.

4.3.2 Модель ранжирования на основе одномерной проекции

Зададим для каждого $x \in C$ числовой параметр $y_x \in \mathbb{R}$, который и будет возвращаемым значением: $f(x) = y_x$. Таким образом множество C проецируется на действительную ось.

Для определения всей совокупности проекций $Y = \{y_x : x \in C\}$ потребуем, чтобы разности $y_v - y_u$ как можно точнее приближали дистанции δ_{uv}. Для этого воспользуемся методом наименьших квадратов:

$$Q(Y) = \sum_{u,v} \frac{w_{uv}}{|R_{uv}|} \sum_{\delta \in R_{uv}} (y_v - y_u - \delta)^2 \to \min_Y,$$

где веса w_{uv} введены для общности постановки задачи; в частности, их можно задать единичными или убывающими по σ_{uv}.

Возможны две постановки задачи - глобальная и локальная:

В глобальной постановке задачи суммирование производится по всем $(u, v) \in C^2$ и значения y также определяются для всех $x \in C$.

В локальной постановке задачи суммирование осуществляется в рамках всех пар $(u, v) \in X^2$ для каждого неупорядоченного множества X.

Значение y_x определены с точностью до константы a, поскольку оптимизируемый критерий $Q(Y)$ зависит только от разности $y_v - y_u$.

Преимущество глобальной постановки в том, что все проекции y_x достаточно вычислить один раз. С другой стороны, согласованное проецирование большого множества точек C на одну ось может привести к более сильным искажениям, чем одномерное проецирование гораздо меньшего множества X.

Теорема 1. Решение оптимизационной задачи удовлетворяет следующей системе уравнений относительно переменных y_x:

$$y_x = \frac{\sum_u \overline{w}_{ux} y_u}{\sum_u \overline{w}_{ux}} + \frac{\sum_u \overline{w}_{ux} \mu_{ux}}{\sum_u \overline{w}_{ux}},$$

$$\overline{w}_{ux} = \frac{w_{xu} + w_{ux}}{2},$$

где всё суммирование производится по $u \in C \backslash x$ для глобальной постановки задачи, и по $u \in X \backslash x$ для локальной постановки задачи.

Следствие 1. Если веса симметричны ($w_{ux} = w_{xu}$), то $\overline{w}_{ux} = w_{ux}$. В нашем случае нет смысла задавать веса несимметричными, поскольку решение зависит только от симметризированных весов.

Система уравнений относительно y_x записана в виде, удобном для применения метода простой итерации. Одной итерации достаточно в тех случаях, когда первая дробь в формуле для y_x оказывается не зависящей от x, следовательно, обращается в константное слагаемое.

Следствие 2. Если $w_{uv} = 1$, то систему уравнений решать не нужно. При этом, в случае локальной постановки задачи, модель ранжирования по одномерной проекции эквивалентна модели средней дистанции.

4.4 Модель ранжирования основанная на градиентном бустинге над решающими деревьями с использованием метаданных о цитируемых статьях

Рассмотрим последовательность групп связанных ссылок $X_D = (X_1, ..., X_k)$ и их целевое ранжирование $(y_1, ..., y_k)$.

Строим алгоритм в виде:

$$f_n(x) = \sum_{i=1}^{n} b_i(x).$$

Пусть построен $f_t(x)$, тогда будем обучать алгоритм b_t на выборке $(x, -\mathcal{L}'(y, f_t(x)))$.

Тогда

$$f_{t+1}(x) = f_t(x) + b_t(x).$$

В задаче ранжирования антиградиент функции потерь $\mathcal{L}(y, f_t(x))$ вычисляется отдельно для каждой группы связанных ссылок X_d уже после сортировки документов по оценкам.

Так как предлагается минимизировать долю неверно отранжированных пар, что не является гладкой функцией, в качестве оптимизационной функции потерь $\mathcal{L}$ будем рассматривать следующие гладкие верхние оценки введенной функции потерь :

$$\mathcal{L}_1(M) = \sum_{v<u} exp(-M);$$

$$\mathcal{L}_2(M) = \sum_{v<u} log(1 + exp(-M)),$$

где $M(f(x), u, v) = f(x_u) - f(x_v)$.

В качестве признакового пространства будем использовать метаданные, описанные в пункте 3.3.

4.4.1 Модифицируем представленную выше модель

Изменим приведенную выше модель с учетом специфики задачи ранжирования. На каждом шаге алгоритма будем двигаться в сторону антиградиента функции потерь умноженного на разницу целевой метрики.

Пусть целевая метрика задается функцией $g(x)$. Тогда под разницей целевой метрики на паре (u, v) будем подразумевать выражения вида:

$$\hat{g}(f(x), u, v) = g(f(x_1), ...f(x_u), ...f(x_v), ...) - g(f(x_1), ...f(x_v), ...f(x_u), ...)$$

Тогда вектор антиградиента оптимизируемой функции по выходам модели будет состоять из объектов вида:

$$\sum_{i<j} -\mathcal{L}'(M(f(x)i, j)) * \hat{g}(f(x), i, j) \ , \forall i \in [0, k_d]$$

В качестве $\mathcal{L}$ будем рассматриваются $\mathcal{L}_1$ и $\mathcal{L}_2$.

5 Анализ работы моделей

При проведение анализа качества работы моделей предлагается рассматривать качество каждой из моделей на всей тестовой воборке. Кроме этого, предлагается проверить гипотезу о том, что на разных объемах подмножества ссылок k_d разные модели могут показывать наилучший результат из предложенных. $g(x)$ - значение ранговой корреляции Кендалла.

Для проверки гипотезы предлагается разделить тестовую выборку на три, в соответствии с количеством цитируемых статей:

	# **цитирований**
Маленькое	$k_d \in [2, 10]$
Среднее	$k_d \in (10, 15]$
Большое	$k_d \in (15, 30]$

Таблица 1: Разделениие тестовой выборки

Для удобного представления результатов приведем общую таблицу (табл. 2) со всеми проверяемыми моделями, а также их названиями, используемыми в сравнениях далее:

Название модели	**Краткое описание алгоритма ранжирования**
PageRank	Основной фактор ранжирования - выход алгоритма PageRank
Weighted PageRank year	Основной фактор ранжирования - выход алгоритма взвешенного PageRank, где веса определяются годом публикации
Weighted PageRank citation count	Основной фактор ранжирования - выход алгоритма взвешенного PageRank, где веса определяются общим количеством цитирований конкретной статьи
Stat model f_0	Учитывать, насколько объект правее других элементов множества
Stat model f_δ	Учитывать, насколько объект правее других элементов конкретной последовательности
Stat model $f_{\delta_{all}}$	Учитывать все дистанции между текущим объектом и другими объектами вне зависимости от их взаимного расположения
Stat model f_μ	Учитывать среднюю дистанцию между объектом и другими элементами последовательности
Stat model f_σ	Учитывать среднюю дистанцию между объектом и другими элементами последовательности с нормировкой на среднеквадратичное отклонение
Year model	Ранжирование по году публикации от старых к новым
Grad model L1	Оптимизация градиентного бустинга на антиградиент $\mathcal{L}_1$
Grad model L2	Оптимизация градиентного бустинга на антиградиент $\mathcal{L}_2$
Grad model L1 delta	Оптимизация градиентного бустинга на антиградиент $\mathcal{L}_1$, домноженный на разность целевой метрики $\hat{g}$
Grad model L2 delta	Оптимизация градиентного бустинга на антиградиент $\mathcal{L}_2$, домноженный на разность целевой метрики $\hat{g}$

Таблица 2: Краткое описание моделей

Модель	$g(x)$
PageRank	0.30940
Weighted PageRank year	0.29439
Weighted PageRank citation count	0.29651
Stat model f_0	0.29292
Stat model f_δ	0.17622
Stat model $f_{\delta_{all}}$	0.21402
Stat model f_μ	0.23105
Stat model f_σ	0.23384
Year model	0.41608
Grad model L1	0.45755
Grad model L2	**0.46025**
Grad model L1 delta	0.40269
Grad model L2 delta	0.40827

Таблица 3: Качество моделей на полном наборе данных

Модель	$g(x)$
PageRank	0.28232
Weighted PageRank year	0.27679
Weighted PageRank citation count	0.27432
Stat model f_0	0.25495
Stat model f_δ	0.12903
Stat model $f_{\delta_{all}}$	0.16620
Stat model f_μ	0.18127
Stat model f_σ	0.18390
Year model	0.38154
Grad model L1	0.42037
Grad model L2	**0.42151**
Grad model L1 delta	0.35946
Grad model L2 delta	0.36445

Таблица 4: Качество моделей на маленьком наборе данных

Модель	$g(x)$
PageRank	0.35904
Weighted PageRank year	0.30032
Weighted PageRank citation count	0.29432
Stat model f_0	0.35290
Stat model f_δ	0.25381
Stat model $f_{\delta_{all}}$	0.29517
Stat model f_μ	0.31785
Stat model f_σ	0.31927
Year model	0.49089
Grad model L1	**0.53994**
Grad model L2	0.53987
Grad model L1 delta	0.48979
Grad model L2 delta	0.49073

Таблица 5: Качество моделей на среднем наборе данных

Модель	$g(x)$
PageRank	0.39308
Weighted PageRank year	0.31203
Weighted PageRank citation count	0.2952
Stat model f_0	0.368854
Stat model f_δ	0.27617
Stat model $f_{\delta_{all}}$	0.31979
Stat model f_μ	0.33174
Stat model f_σ	0.33471
Year model	0.50069
Grad model L1	0.55063
Grad model L2	**0.55553**
Grad model L1 delta	0.49023
Grad model L2 delta	0.51220

Таблица 6: Качество моделей на большом наборе данных

Из табл. 3, табл. 4, табл. 5, табл. 6 можем сделать вывод, что лучшим подходом оказывается использование LightGBM, оптимизирующего функционал $\mathcal{L}_2$.

На среднем наборе данных табл. 5 - лучшее качество достигается при использовании модели **Grad model L1**. Однако разницу в качестве у двух лидеров **Grad model L1** и **Grad model L2** можно считать незначительной.

Стоит также заметить, что все перечисленные модели могут быть улучшены путем расширения доступного корпуса для обучения / построения графа цитирований.

Кроме того, из проведенных экспериментов можно сделать вывод, что в данной задаче модели ранжирования, основанные на машинном обучении, показывают более высокое качество в сравнении с алгоритмами, основанными на графе цитирований и статистиках совместного распределения цитирований в корпусе документов.

Отдельно стоит рассмотреть модель, использующую год публикации в качестве фактора для ранжирования. Стоит заметить, что ее качество хоть и уступает моделям машинного обучения на всех вариантах набора данных, тем не менее оказывается выше, чем у любых других моделей, основанных на графах цитирований. При этом, ее важным отличием является отсутствие необходимости в каком-либо процессе обучения. При ее использовании можно избежать проблемы холодного старта, которая может негативно влиять на качество других моделей.

6 Заключение

В данной работе исследованы несколько подходов к решению задачи построения модели ранжирования ссылок в системе полуавтоматического реферирования.

Рассмотрены алгоритмы, основанные на графе цитирований, статистиках совместного распределения статей в корпусе документов и моделях машинного обучения.

При анализе качества моделей на выбранной коллекции лучшим оказался алгоритм, основанный на оптимизации специально подобранной функции потерь градиентным бустингом над решающими деревьями.

Также в ходе анализа было показано, что модели, основанные на графе цитирования и статистиках совместного распределения, в среднем достаточно сильно уступают по качеству моделям машинного обучения.

Список литературы

[1] Ke G. et al. Lightgbm: A highly efficient gradient boosting decision tree //Advances in neural information processing systems. – 2017. – Т. 30.

[2] Chen T., Guestrin C. Xgboost: A scalable tree boosting system //Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. – 2016. – С. 785-794.

[3] Lo K. et al. S2ORC: The semantic scholar open research corpus //arXiv preprint arXiv:1911.02782. – 2019.

[4] Yu P. S., Li X., Liu B. On the temporal dimension of search //Proceedings of the 13th international World Wide Web conference on Alternate track papers posters. – 2004. – С. 448-449.

[5] Chen P. et al. Finding scientific gems with Google's PageRank algorithm //Journal of informetrics. – 2007. – Т. 1. – №. 1. – С. 8-15.

[6] Ding Y. Applying weighted PageRank to author citation networks //Journal of the American Society for Information Science and Technology. – 2011. – Т. 62. – №. 2. – С. 236-245.

[7] Xing W., Ghorbani A. Weighted pagerank algorithm //Proceedings. Second Annual Conference on Communication Networks and Services Research, 2004. – IEEE, 2004. – С. 305-314.

[8] Yan E., Ding Y. Discovering author impact: A PageRank perspective //Information processing management. – 2011. – Т. 47. – №. 1. – С. 125-134.

[9] Bird S. et al. The ACL Anthology Reference Corpus: A Reference Dataset for Bibliographic Research in Computational Linguistics //LREC. – 2008.

[10] Li H. A short introduction to learning to rank //IEICE TRANSACTIONS on Information and Systems. – 2011. – Т. 94. – №. 10. – С. 1854-1862.

[11] Burges C. J. C. From ranknet to lambdarank to lambdamart: An overview //Learning. – 2010. – Т. 11. – №. 23-581. – С. 81.