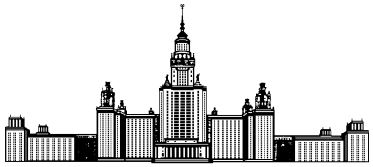
Московский государственный университет имени М. В. Ломоносова



Факультет Вычислительной Математики и Кибернетики

Кафедра Математических Методов Прогнозирования

Крыжановская Светлана Юрьевна

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

«Технология полуавтоматической суммаризации тематических подборок научных статей»

Научный руководитель:

д.ф-м.н., профессор

Воронцов Константин Вячеславович

Москва, 2022

Содержание

1 Введение **3**

1.1 Структура работы 4

2 Постановка задачи **5**

2.1 Определения и обозначения 6

2.2 Обзор литературы 7

3 Методогия **9**

3.1 Построение сценария обзора 9

3.2 Суфлёр, возвращающий фразы аннотации 9

3.3 Суфлёр, возвращающий обобщающие фразы 10

3.4 Суфлёр, возвращающий цитирующие фразы 10

3.5 Суфлёр, возвращающий цитируемые фразы 12

4 Сбор данных **14**

5 Оценка качества **19**

5.1 Метрики качества суммаризации 19

5.2 Генерация реферата 20

6 Эксперименты **22**

6.1 Суфлёр, возвращающий цитирующие фразы 22

6.2 Суфлёр, возвращающий цитируемые фразы 24

6.3 Оценка качества всех суфлёров 25

7 Ресурс «Мастерская знаний» **27**

8 Заключение **28**

Список литературы **30**

Аннотация

Целью автоматической суммаризации является синтез краткого изложения одного или нескольких текстовых документов для быстрого упрощённого понимания ключевых идей. В данной работе рассматривается полуавтоматическая суммаризация коллекций научных публикаций, которая имеет другую конечную цель — помочь пользователю реализовать свой авторский замысел. Обзоры, написанные по одной и той же подборке разными авторами и/или для разных целей (диссертация, отчёт, учебник) могут различаться существенно. В таких случаях пользователю нужен не автомат, а рекомендательный сервис, который помогал бы ему составлять текст обзора фраза за фразой, полностью контролируя процесс. Такие системы называются автоматизированной авторской суммаризацией текстов (machine aided human summarization, MAHS).

В данной работе предлагается процесс MAHS для тематических коллекций научных публикаций, основанный на решении серии задач машинного обучения. В основе предложенной системы лежит механизм ранжирования фраз с помощью нескольких функций (суфлёров), отражающих различные аспекты статьи. В работе рассматривается следующий набор суфлёров: обобщающий суфлёр; суфлёр, выдающий фразы аннотации; суфлёр, показывающий, как на данную статью ссылаются другие авторы; суфлёр, подсказывающий, какие предложения статьи наиболее вероятно используются для написания цитат. Для обучения каждого суфлёра формируется выборка и ставится отдельная задача машинного обучения.

Для тестирования конвейера задач в совокупности предлагается метод построения обучающей выборки из текстов самих научных статей, а также методика оценки качества работы системы, не требующая взаимодействия пользователя с системой.

В экспериментальной части работы исследуется работа каждого суфлёра по отдельности, а также работа всей системы. Оценки качества результирующей модели показывают её преимущество в сравнении с полностью автоматическими методами.

1 Введение

С каждым годом объёмы доступной текстовой информации растут, и их обработка требует все больше человеческих ресурсов. В такой ситуации задача суммаризации текста становится все более актуальна. Алгоритмы автоматической суммаризации позволяют быстро оценить релевантность текста, значительно сократив время на ознакомление с полным содержанием.

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

Системы суммаризации особенно актуальны для учёных и экспертов, которые вынуждены тратить много времени на чтение научных публикаций. В данной работе рассматривается задача полуавтоматической суммаризации, цель которой помочь пользователям написать авторский обзор (реферат, дайджест) по заданной тематической подборке. Отличие данной ситуации от автоматической суммаризации в том, что пользователь реализует свой авторский замысел. Написание обзорной главы диссертации, отчёта по НИР или учебника по одной и той же подборке научных публикаций наверняка приведут к разным результатам. Разные авторы сделают это по-разному, расставляя смысловые акценты исходя из своих целей. Полагаться на алгоритмы автоматической суммаризации в таких случаях едва ли целесообразно. В таких случаях пользователю нужен не готовый текст, а рекомендательный сервис, выполняющий рутинные операции информационного поиска, например, подбирающий релевантные фразы.

Автоматизированная авторская суммаризация (machine aided human summarization, MAHS) – *система подсказок, позволяющая пользователю самому составлять реферат фраза за фразой, оставаясь его главным автором.*

Научная новизна данной работы состоит в декомпозиции процесса MAHS на серию задач машинного обучения, разработке методик сбора данных и оценки качества такой системы. Особенность предложенного подхода заключается в том, что система позволяет пользователю выбирать фразы для продолжения обзора, исходя из разных аспектов. Таким образом, пользователь может сам задавать сценарий реферата в зависимости от своих целей.

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

1.1 Структура работы

Исследование включает в себя восемь глав, включая введение и заключение, а также и список литературы.

Вторая глава содержит формальную постановку задачи, описание ее декомпозиции на серию задач машинного обучения, а также обзор существующих методов суммаризации и оценку их применимости в рамках поставленной задачи.

В третьей главе ставятся задачи машинного обучения для каждого из блоков системы, описываются подходы для их решения. Две основные задачи – ранжирование документов в подборке и ранжирование фраз для каждого документе в порядке релевантности.

В четвёртой главе приводится обзор существующих датасетов для суммаризации научных статей, а также предлагается методика сбора данных из подборок статей и их рефератов, получаемых из обзорных частей научных публикаций.

В пятой главе рассматриваются основные метрики качества суммаризации и приводится алгоритм автоматической генерации реферата, что делает возможным оценку качества предложенной системы без участия человека.

В шестой главе представлены эксперименты с анализом полученных результатов по оценке качества всей системы и её отдельных блоков.

В седьмой главе приводится прототип системы в поисково-рекомендательной системе «Мастерская знаний».

В заключении подводятся основные итоги исследования и определяются перспективы его дальнейшего развития.

2 Постановка задачи

Дана выборка из нескольких научных статей (подборка, тематическая коллекция), привязанных к некоторому графу цитирований. Требуется построить систему ранжирования фраз, отражающих различные аспекты всех статей.

Предлагается процесс MAHS, состоящий из двух этапов:

1. Ранжирование документов в подборке в том порядке, в котором они будут упоминаться в реферате (построение сценария обзора).

2. Ранжирование фраз-подсказок в порядке релевантности для каждого документа из подборки, из которых пользователь может выбирать фразы для продолжения своего обзора.

Существует множество разумных способов отбора и ранжирования фраз. Они реализуются в системе как функции ранжирования, называемые суфлёрами. Для обучения каждого суфлёра ставится отдельная задача машинного обучения. В данной работе рассматриваются следующие суфлёры:

- Суфлёр, возвращающий фразы аннотации (abstract prompter);

- Суфлёр, возвращающий фразы статьи, которые описывают ее наилучшим образом (overview prompter);

- Суфлёр, показывающий как на данную статью ссылаются другие авторы (citance prompter);

- Суфлёр, возвращающий фразы из статьи, близкие к цитирующим фразам (reference prompter) .

В рамках исследования ставятся следующие задачи:

1. Постановка задач машинного для обучения суфлёров и их решение.

2. Формирование обучающей выборки для задачи суммаризации подборок статей.

3. Исследование методик по оценке качества предложенной системы MAHS из нескольких суфлёров.

Процесс создания реферата с помощью предложенной системы MAHS представлен на Рис. 1.

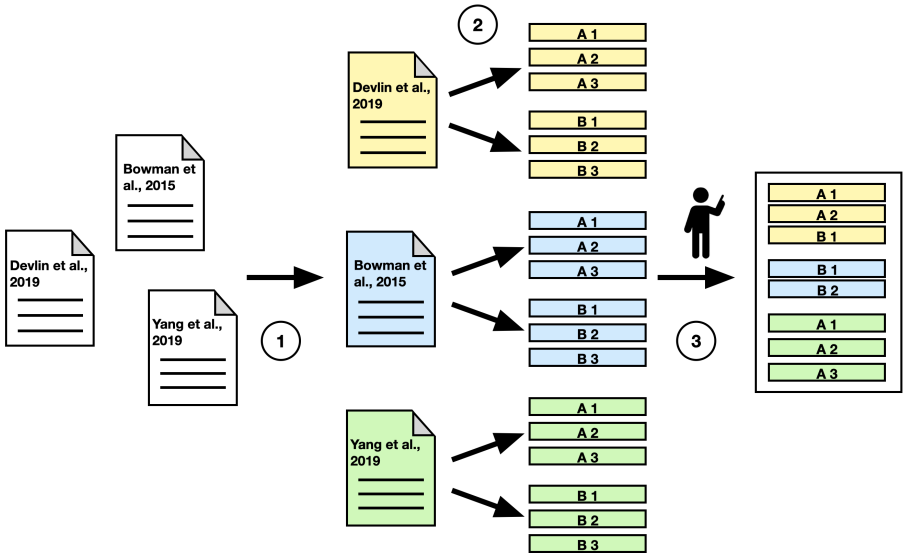


Рис. 1: Процесс создания реферата для подборки статей: 1) построение сценария обзора; 2) ранжирование фраз-подсказок с помощью различных суфлёров (на рисунке суфлёры обозначены буквами A и B) для каждого документа; 3) пользователь последовательно просматривает документы и отбирает фразы для реферата.

2.1 Определения и обозначения

Суфлёр (prompter) *– функция ранжирования фраз, отражающих некоторый аспект статьи. Наличие нескольких суфлёров позволяет пользователю переключаться между ними и самому задавать сценарий реферата.*

Граф цитирований (citation graph) *- граф, каждая вершина которого представляет документ в коллекции, а каждое ребро направлено от одного документа к другому, который он цитирует.*

Цитирующий фрагмент (citation span, citance) *– одно или несколько предложений в цитирующей статье (citing paper, RP), которые содержат цитату и обсуждение цитируемой статьи (reference paper, RP) авторами. Текстовый фрагмент цитаты может состоять только из предложения / части предложения, содержащего саму цитату, а также включать предложения / части предложений до или после самой цитаты.*

Цитируемый фрагмент, ссылочный промежуток (reference span) *– одно или несколько предложений в цитируемой статье, семантически близкие к цитирующему фрагменту.*

Суммаризация на основе цитирований (citation based summarization) *– суммаризация научной публикации, которая состоит из цитирующих / цитируемых фрагментов на данную публикацию.*

2.2 Обзор литературы

Существующие алгоритмы суммаризации делятся на экстрактивные (extractive) и абстрактивные (abstractive). Экстрактивные методы выделяют самые информативные фрагменты текста (как правило, предложения) и составляют из них краткое содержание. Недостаток экстрактивных подходов в том, что составленный таким образом реферат может получиться не связным и трудно читаемым. Абстрактивные методы генерируют новый текст, отражающий основные идеи исходного. В сравнении с экстрактивными методами, текст получается более связный, однако общий смысл может быть искажён. Стоит отметить, что несмотря на то что абстрактивные модели более приближены к поведению человека при написании реферата, большинство современных систем автоматической суммаризации работает на основе экстрактивных подходов.

Среди экстрактивных методов можно выделить методы обучения с учителем (supervised) и без (unsupervised). Обычно, суть unsupervised методов заключается в том, что все фрагменты статьи некоторым образом получают оценку важности и реферат составляется из фрагментов с самой большой важностью. Так, метод LSA [18] для определения веса каждого из предложений использует SVD (Singular Value Decomposition) разложение. KL-Sum [11] строит реферат, распределение слов в котором больше всего похоже на распределение слов в исходном тексте. Графовые методы основаны на представлении текста в виде графа, используя схожесть между предложениями. При этом вес каждого предложения оценивается исходя из структуры полученного графа. Одним из таких подходов является алгоритм TextRank [16]. Supervised подходы рассматривают задачу экстрактивной суммаризации как задачу бинарной классификации предложений. Среди supervised подходов наилучшее

качество показывают модели, основанные на нейронных сетях. Это связано с возможностью предобучения таких моделей на большом корпусе текстов и дальнейшем использовании предобученных моделей для дообучения на конкретную задачу. Одной из state-of-the-art моделей для задачи суммаризации является модель BertSum [13], основанная на модели BERT [5].

Все современные методы абстрактивной суммаризации основаны на моделях нейронных сетей, имеющих encoder-decoder архитектуру. В такой архитектуре кодировщик (encoder) преобразует входной документ в скрытое представление, на основе которого декодировщик (decoder) генерирует итоговый реферат. Особенный прогресс абстрактивные системы получили с появлением модели Transformer [1]. Примерами Transformer-моделей, показывающих state-of-the-art качество среди абстрактивных методов, являются BART [3], PEGASUS [19] и T5 [10]. Однако, Transformer-модели могут принимать на вход текст не более чем из 15-20 предложений, что делает их применение к длинным текстам проблематичным.

Особенностью научных публикаций является наличие цитирующих статей, которые отражают влияние исследуемой статьи на научное сообщество. Методы суммаризации научных публикаций на основе цитирований ([15]) используют эту информацию для составления реферата, извлекая наиболее релевантные текстовый фрагменты цитат (citances) из цитирующих статей. Однако, текстовые фрагменты цитат помимо обсуждения цитируемой статьи часто содержат много нерелевантной информации. Для решения этой проблемы в [1] и [8] вместо текстовых фрагментов цитат определяется набор цитируемых участков текста (reference spans), и на основе найденных фрагментов составляется реферат. Таким образом, хотя резюме состоит из фрагментов, содержащихся в исходной статье, оно отражает мнение исследовательского сообщества. Кроме того, большинство научной литературы имеет чётко определённую структуру разделов. Эта информация может быть так же полезна для составления реферата. Так, в [9] обучается модель сегментации документа, после чего определяется вес каждого раздела, определяющий его вклад в итоговый реферат.

Большинство существующих подходов нацелены на полностью автоматическое создание реферата. Среди полуавтоматических систем, можно выделить систему SAAR [14], позволяющую человеку редактировать автоматизированный реферат

(Human Aided Machine Summarization). Такая система генерирует реферат, показывает его пользователю, и, если он не одобряет его, генерирует новый реферат в соответствии с обратной связью.

3 Методогия

3.1 Построение сценария обзора

Дана подборка статей, требуется ранжировать их в том порядке, в котором они будут упоминаться в обзоре.

Обучающая выборка может быть составлена автоматически по большой коллекции научных статей. Каждая статья порождает обучающий объект, в котором роль подборки выполняет список литературы, а обучающим ранжированием является последовательность ссылок в обзорной части статьи.

Информативными признаками являются год публикации, её цитируемость, цитируемость её авторов, семантическая близость публикации к обзору, к его локальному контексту, и т.д. Подробно задача построения обзора рассматривается в [23]. В качестве бейзлайна автор рассматривает ранжирование по году публикации. Также в работе предлагается использовать метод машинного обучения на основе бустинга.

В данной работе будем использовать ранжирование документов по году публикации.

3.2 Суфлёр, возвращающий фразы аннотации

Аннотация отражает основные идеи статьи с точки зрения автора. Кроме того, в [9] было показано, что большинство рефератов, созданных человеком, содержат большое количество терминов из аннотации.

Большинство научных публикаций имеют аннотацию, которая, как правило, легко выделяется с помощью специальных парсеров. В качестве суфлёра аннотации предлагается использовать простейшую необучаемую функцию, которая выдаёт фразы из аннотации статьи в их исходном порядке.

3.3 Суфлёр, возвращающий обобщающие фразы

Для построения суфлёра, возвращающего фразы статьи, которые описывают ее наилучшим образом, предлагается применить классические методы экстрактивной суммаризации.

Алгоритмы экстрактивной суммаризации обычно работают в 3 этапа:

- Построение внутреннего представления для предложений документа
- Оценка важности каждого предложения (ранжирование).
- Отбор предложений для реферата на основе оценок важности.

В случае построения суфлёра предлагается ранжировать предложения согласно оценкам важности экстрактивного метода. Стоит отметить, что научные публикации, как правило, представляют собой довольно длинные документы, что накладывает дополнительные ограничения на используемые методы (например, модели суммаризации с transformer-based архитектурой имеют ограничение в 512 токенов). В данной работе были исследованы методы TextRank, KL-Sum и LSA.

3.4 Суфлёр, возвращающий цитирующие фразы

Даны тексты статьи и цитирующих её статей с выделенными цитатами. Для каждой цитаты требуется выделить цитирующий фрагмент в цитирующей статье и отранжировать полученные фразы в порядке релевантности.

Сложность выделения цитирующих фрагментов заключается в том, что рядом может быть несколько цитат, цитирующие фрагменты которых могут не пересекаться или совпадать. В итоге, цитирующий фрагмент может содержать часть предложения, одно предложение или несколько предложений. Примеры цитирующих фрагментов представлены на Рис. 2.

Other work includes Li and Li (2002) who propose a bilingual bootstrapping method to learn a translation disambiguation WSD model, and *Diab (2004) who exploited large amounts of automatically generated noisy parallel data to learn WSD models in an unsupervised bootstrapping scheme.*
Similarly, Belinkov et al. (2017) and Belinkov et al. (2018) give detailed analyses of both encoder and decoder's learned knowledge about part-of-speech and semantic tags at different layers.
Chang et al. (2010) proposed a discriminative linear model where alignments are treated as hidden structures, and the sentence-level semantic relation is derived based on the best latent alignment structure. They formulated the problem of predicting the best hidden structure as an Integer Linear Programming problem, where domain knowledge is encoded as constraints.

Рис. 2: Примеры цитирований. Цитаты выделены красным цветом, а цитирующие фрагменты – курсивом.

Для обучения моделей машинного обучения для выделения цитирующих фрагментов требуется размеченная обучающая выборка вида <контекст цитаты, цитирующий фрагмент>. При этом задачу локализации цитирующего фрагмента можно рассматривать, как задачу поиска ответа в заданном тексте, отвечающего на запрос. В данном случае запросом может выступать сама цитата, а в качестве текста целесообразно рассматривать окно, содержащее предложение с цитатой и предложения справа и слева в рамках параграфа.

Бейзлайн. В качестве бейзлайна предлагается брать одно предложение, содержащее цитату.

Модель BERT. Модель BERT успешно зарекомендовала себя для задач вопросно-ответного поиска [5]. В текущей задаче предлагается использовать модель SciBERT [4], предобученную на 1.14M научных статей. Для решения задачи вопросно-ответного поиска, на вход модели подаются сконкатенированные токенизированные представления для запроса и текста, разделённые специальным токеном <SEP>, как показано на Рис. 3. Пусть $T'_i \in \mathcal{R}^h$ – последнее внутренне представление модели для токена i из текста для поиска фрагмента. Тогда вероятность того, что токен i является началом нужного фрагмента определяется как $P_{start}(i) = \frac{\exp S^T \cdot T'_i}{\sum_j \exp S^T \cdot T'_j}$,

где $S \in \mathcal{R}^h$ – обучаемый вектор проекции. Аналогично определяется вероятность $P_{end}(i)$ того, что токен i является концом фрагмента ответа. В качестве ответа выбирается фрагмент от i-го токена до i'-го токена, такой что $i' \geq i+15$ и $P_{start}(i) \cdot P_{end}(i')$ максимально.

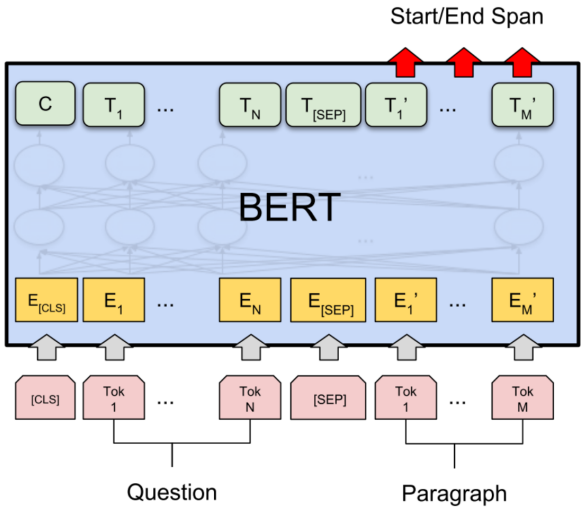


Рис. 3: Обучение модели BERT для задач вопросно-ответного поиска.

Для ранжирования фраз из цитирующих фрагментов предлагается использовать классические методы экстрактивной суммаризации.

3.5 Суфлёр, возвращающий цитируемые фразы

Дан текст статьи и цитирующих её статей, в которых выделены цитаты на данную статью. Требуется построить функцию, ранжирующую предложения из самой статьи, близкие к локальным контекстам ссылок в цитирующих статьях. В качестве локального контекста цитаты достаточно брать предложение, содержащее цитату. Далее будем называть его цитирующим предложением.

Пусть $S = \{s_i\}_{i=1}^N$ – множество предложений статьи, $C = \{c_j\}_{j=1}^M$ – множество цитирующих предложений, $score(s, c)$– модель, оценивающая семантическую близость s и c.

Алгоритм 1 Алгоритм работы суфлёра

Вход: множество предложений статьи S, множество цитирующих предложений C

Выход: $s_{(1)} \succ s_{(2)} \succ \cdots \succ s_{(N)}$ – предложения статьи в порядке релевантности

для всех $s \in S$

$a(s) = \sum_{c \in C} score(s, c)$

$s_{(i)} \succ s_{(j)}$ если $a(s_{(i)}) > a(s_{(j)})$

Далее рассмотрим модели для оценки семантической близости $score(s, c)$.

Для обучения моделей машинного обучения требуется размеченная обучающая выборка вида <цитируемый фрагмент, цитирующий фрагмент>. Исходную задачу можно рассматривать как задачу бинарной классификации для пары предложений (s, c), а именно для каждой пары необходимо оценить вероятность того, что s входит в цитируемый фрагмент для c.

Бейзлайн. В качестве необучаемой функции $score(s, c)$ предлагается взять косинусное расстояние между эмбеддингами предложений s и c, полученными из языковой модели SBERT [20].

Классические модели машинного обучения.

Для решения задачи на фиксированном наборе признаков можно использовать модели градиентного бустинга, а также модель полносвязной нейронной сети (Multilayer Perceptron). Подробно использование классических моделей рассматривается в [23]. Автор выделяет следующий набор информативных признаков для пары (s, c):

- Косинусное расстояние между: tf-idf векторами, LDA эмбеддингами [6], усреднёнными эмбедингами fasttext [2];
- Количество общих n-грамм;
- $ROUGE_1$, $ROUGE_2$, $ROUGE_L$ [12];
- Позиционные признаки: позиция предложения в статье, позиция предложения в текущей секции статьи, порядковый номер текущей секции.

Модель BERT. Предлагается использовать модель SciBERT [4], предобученную на 1.14M научных статей. Для классификации пары (s, c) на вход модели подаются

сконкатенированные токенизированные представления для s и c, разделённые специальным токеном <SEP>. Пусть $C \in \mathcal{R}^h$ – последнее внутреннее представление специального токена <CLS>. Тогда вероятность того, что s входит в цитируемый промежуток c равна $sigmoid(W^T \cdot C)$, где $W \in \mathcal{R}^h$ - обучаемый вектор проекции.

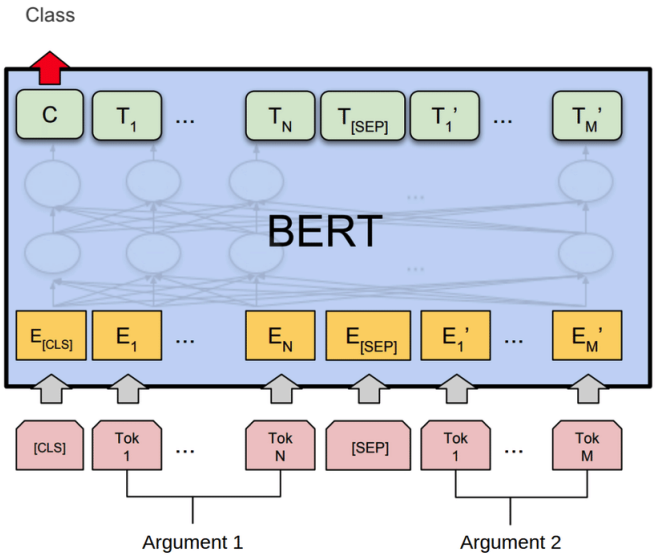


Рис. 4: Обучение модели BERT для классификации пары предложений.

4 Сбор данных

На текущий момент все существующие датасеты ориентированы на суммаризацию одной научной статьи. Так, в рамках конкурса CL-SciSumm 2018 [7] рассматривается задача суммаризации статей из области компьютерной лингвистики. Помимо рефератов, составленных экспертами в области по полному тексту статьи, предлагаются рефераты, составленные из внешних цитат. Для расширения датасета [22] предлагают упростить процесс разметки, предлагая экспертам только аннотацию и предложения, содержащие внешние цитаты. Полученный таким образом датасет ScisummNet состоит из 1000 рефератов в области компьютерной лингвистики. Также стоит отметить наборы данных LaySumm и LongSumm [17]. LaySumm предоставляет рефераты, написанные для нетехнической аудитории, а LongSumm ориентирован на написание длинных рефератов. При этом авторы LongSumm предлагают два типа рефератов: экстрактивные, основанные на видео-докладах с конференций, и абстрактивные, полученные из постов научных блогов. Стоит заметить, что такое обилие

датасетов с разными целями и структурами рефератов подтверждает актуальность текущего исследования в поиске универсального решения.

В данной работе предлагается собрать новый набор данных из рефератов и подборок статей, используя обзорные части статей. В роли «эталонного» реферата будем использовать обзорную часть статьи, а в роли соответствующей данному реферату подборки – документы, которые упоминаются в обзорной части. Кроме того, для каждого документа в подборке необходимо найти цитирующие статьи.

Для сбора данных предлагается использовать корпус S2ORC (The Semantic Scholar Open Research Corpus) [21]. S2ORC состоит из 136M научных публикаций, охватывающих множество различных академических дисциплин, среди которых 12.7M представлены с полным текстом. Кроме того, статьи объединены в граф цитирований, содержащий 467M рёбер.

Корпус S2ORC обладает следующими характеристиками:

- Наличие метаданных: заголовок, имена авторов, год публикации, журнал, а также списки цитирующих и цитирующихся статей.
- Определены аннотации.
- Текст разделён на параграфы, часть из которых имеет заголовок.
- Определены внутритекстовые цитаты, ссылки на таблицы и рисунки.
- Определены статьи библиографического списка, таблицы и рисунки. Элементы библиографии связаны со статьями из корпуса.
- Внутритекстовые цитаты связаны с элементами библиографии, ссылки на таблицы и рисунки – с соответствующими элементами рисунков/ таблиц. Пример связывания представлен на Рис. 5.

Рис. 5: S2ORC: структура корпуса.

Из-за технических ограничений для сбора данных была выбрана небольшая подвыборка коллекции S2ORC. Итоговый набор данных был получен согласно следующему алгоритму:

1. Выгружена подвыборка S2ORC статей ACL (Association for Computational Linguistics) из области компьютерной лингвистики (43k публикаций).

2. Среди статей ACL выбраны, которые содержат раздел «Related Work» (13k публикаций).

3. Среди статей с разделом «Related Work» выбраны те, для которых все статьи, упоминающиеся в раздел «Related Work», также содержатся в подвыборке ACL (450+ публикаций).

4. Из каждой статьи сформирован элемент выборки: <подборка, «эталонный» реферат>, где в качестве подборки выступают статьи, упоминающиеся в разделе «Related Work», а в качестве «эталонного» реферата – раздел «Related Work».

   Процесс формирования элемента выборки проиллюстрирован на Рис. 6, а пример полученного реферата и соответствующей подборки представлен на Рис. 9.

5. Оставлены элементы, в которых подборка состоят хотя бы из 2 документов. (350+ публикаций).

6. Для каждого документа в подборке найдены цитирующие документы в подвыборке ACL.

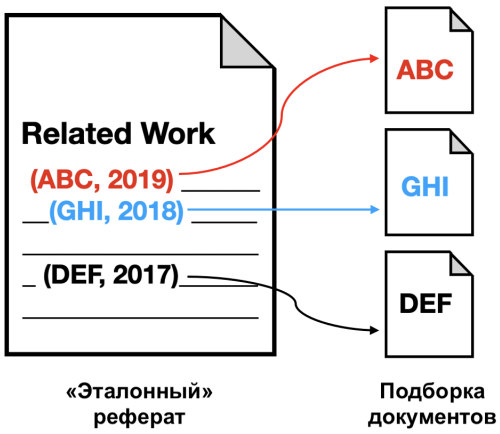


Рис. 6: Формирование элемента выборки.

Из собранных данных для тестовой подвыборки были вручную отобраны 35 рефератов ($\approx 10\%$). Полученные подборки состоят из 2 – 14 документов, распределение числа документов в подборках представлено на Рис. 7. Длина рефератов варьируется от 3 до 38 предложений, распределение длин рефератов представлено на Рис. 8. Также отметим, что для статей из подвыборки ACL примерно в половине случаев цитирующие статьи также находятся в подвыборке ACL. Детальные характеристики собранного датасета представлены в Таб. 1.

	Train	Test
# примеров в выборке	322	35
# документов	1213	221
# цитирующих документов	21725	8999
# документов в подборке	2 – 14	2 – 14
# цитирующих документов в подборке	4 – 3368	39 – 1743
% найденных цитирующих документов	0.46	0.46

Таблица 1: Характеристики собранного датасета для суммаризации подборок научных статей.

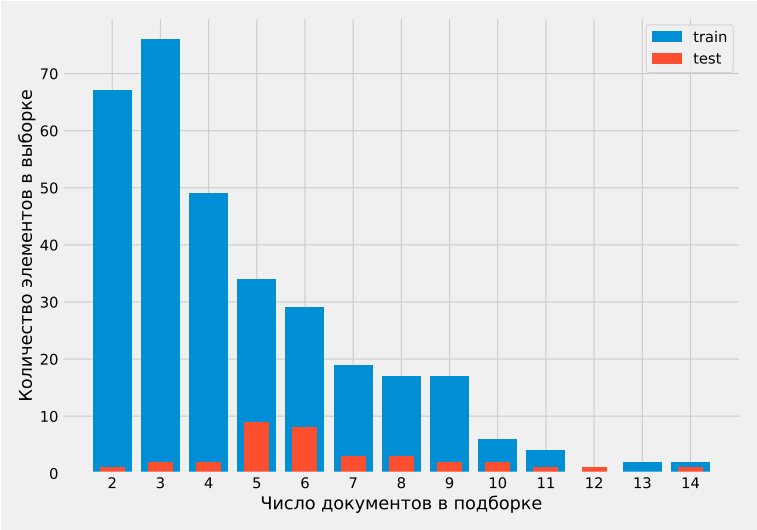


Рис. 7: Распределение числа документов в подборках.

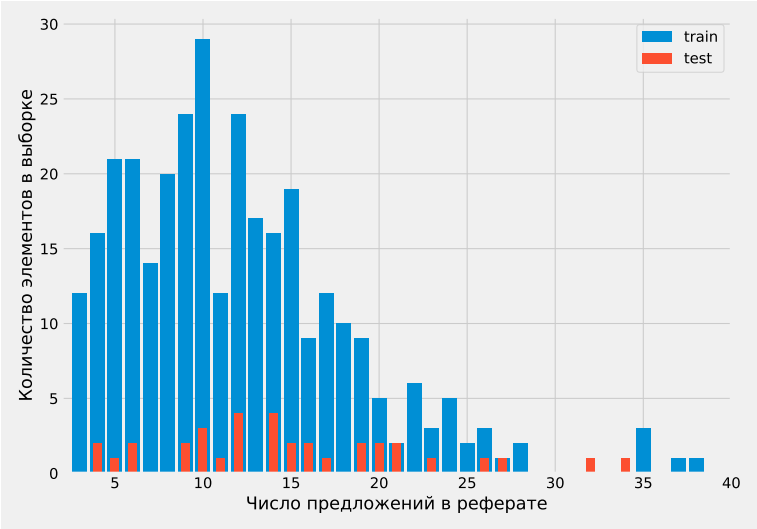


Рис. 8: Распределение длин рефератов (в предложениях).

Заметим, что с помощью предложенного метода сбора данных датасет может быть легко расширен новыми обзорными разделами. Например, обзорными разделами можно также считать разделы «Introduction» и «Background». Кроме того, автоматизировать сбор данных можно с помощью выделения обзорных частей методами машинного обучения.

Most of the systems dedicated to ABSA use machine learning algorithms such as SVMs (Wagner et al., 2014; Kiritchenko et al., 2014) , or CRFs (Toh and Wang, 2014; Hamdan et al., 2015) , which are often combined with semantic lexical information, n-gram models, and sometimes more fine-grained syntactic or semantic information. For example, (Kumar et al., 2016) proposed a very efficient system on different languages of SemEval2016. The system use information extracted from dependency graphs and distributional thesaurus learned on the different domains and languages of the challenge. Deep Learning methods are also emerging: for example, (Ruder et al., 2016) proposed a method using multiple filters CNNs and obtained competitive results on both polarity and aspect detection tasks. However, ABSA datasets are very costly to annotate by humans, and they are usually small, which is a problem for Deep Learning supervised methods.

Рис. 9: Пример реферата, полученного из обзорной главы. Цитаты на документы из соответствующей подборки выделены красным цветом.

5 Оценка качества

5.1 Метрики качества суммаризации

Для автоматической оценки качества суммаризации чаще всего используется семейство метрик ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [12]. Все они определяют меру сходства автоматически сгенерированного реферата с эталонным. Чаще всего используются метрики $ROUGE_1$, $ROUGE_2$ и $ROUGE_L$:

$$ROUGE_N(p) = \frac{Count_{match}(n\text{-}gramm)}{Count_{ref}(n\text{-}gramm)}, \quad ROUGE_N(r) = \frac{Count_{match}(n\text{-}gramm)}{Count_{hyp}(n\text{-}gramm)},$$

$$ROUGE_N(f1) = 2\frac{ROUGE_N(p)ROUGE_N(r)}{ROUGE_N(p) + ROUGE_N(r)},$$

где $Count_{match}(n\text{-}gramm)$ – кол-во n-грамм, встречающихся и в автоматическом реферате, и в эталонном, $Count_{ref}(n\text{-}gramm)$ – кол-во n-грамм в эталонном реферате, $Count_{hyp}(n\text{-}gramm)$ – кол-во n-грамм в автоматическом реферате.

$$ROUGE_L = \frac{(1+\beta^2)R_{LCS}P_{LCS}}{R_{LCS} + \beta^2 P_{LCS}}, \quad R_{LCS} = \frac{LCS(X,Y)}{|X|}, \quad P_{LCS} = \frac{LCS(X,Y)}{|Y|},$$

где X – последовательность слов эталонного реферата, Y – последовательность слов автоматического реферата, $LCS(X, Y)$ – длина наибольшей общей подпоследовательности между X и Y, $\beta = P_{LCS}/R_{LCS}$.

5.2 Генерация реферата

Дан сценарий обзора и ранжированные списки фраз от всех суфлёров для каждого документа в коллекции, а также «эталонный реферат». Необходимо автоматически генерировать конечный реферат.

Аналогично тому, как пользователь составляет реферат, будем выбирать «наилучшие» фразы, последовательно просматривая документы. Заметим, что пользователь может не выбрать «наилучшую» с точки зрения суфлёра фразу, а последовательно просматривать выдачу до нахождения подходящей фразы. Легко провести аналогию с пользователями, выбирающими нужную страницу в поисковой выдаче браузера. Далее будем считать, что среди первых k фраз в выдаче пользователь сможет найти подходящую.

Введём обозначения:

$D = \{d_i\}_{i=1}^{N}$– подборка документов,

$d_{(1)} \succ d_{(2)} \succ \cdots \succ d_{(N)}$– сценарий реферата,

$\{P_j\}_{j=1}^{M}$– набор функций-суфлёров,

$P_j(d_{(i)}) = \left[s_j^1, s_j^2, \ldots, s_j^{N_j}\right]$ – упорядоченный список фраз j-го суфлёра для i-го документа

$top_k\left[P_j(d_{(i)})\right] = \left[s_j^1, s_j^2, \ldots, s_j^k\right]$ – первые k фраз из выдачи.

Алгоритм 2 Алгоритм генерации реферата.

Вход: подборка документов $D = \{d_i\}_{i=1}^{N}$, сценарий реферата $d_{(1)} \succ d_{(2)} \succ \cdots \succ d_{(N)}$, набор функций-суфлёров $\{P_k\}_{k=1}^{M}$, S_{gold} – «эталонный» реферат, k – число top фраз в выдаче для выбора следующей фразы

Выход: S – набор фраз для реферата

$S = \emptyset$

$i = 1$

пока $i \leq N$ ▷ Рассматриваем документ $d_{(i)}$

$\hat{S} = \bigcup_{j=1}^{M} top_k \left[P_j(d_{(i)})\right]$ ▷ top-k фраз из выдачи всех суфлёров

если $\hat{S} = \emptyset$ **то**

$i = i + 1$ ▷ Переходим к следующему документу

иначе

$s_{best} = \{s \in \hat{S} \;\big|\; ROUGE(S \cup \{s_{best}\}, S_{gold}) \geq ROUGE(S \cup \{s\}, S_{gold}) \;\; \forall \;\; s \in \hat{S}\}$

$j_{best} = \{j \in [1, \ldots, M] \;\big|\; s_{best} \in P_{j_{best}}(d_{(i)}) \;\}$

если $ROUGE(S \cup \{s_{best}\}, S_{gold}) > ROUGE(S, S_{gold})$ **то**

Добавить s_{best} в S

Удалить s_{best} из выдачи $P_{j_{best}}(d_{(i)})$

иначе

$i = i + 1$ ▷ Переходим к следующему документу

Таким образом, становится возможна автоматическая оценка качества системы на размеченной выборке. Также можно оценить вклад каждого суфлёра как количество фраз, которые вошли в реферат.

6 Эксперименты

6.1 Суфлёр, возвращающий цитирующие фразы

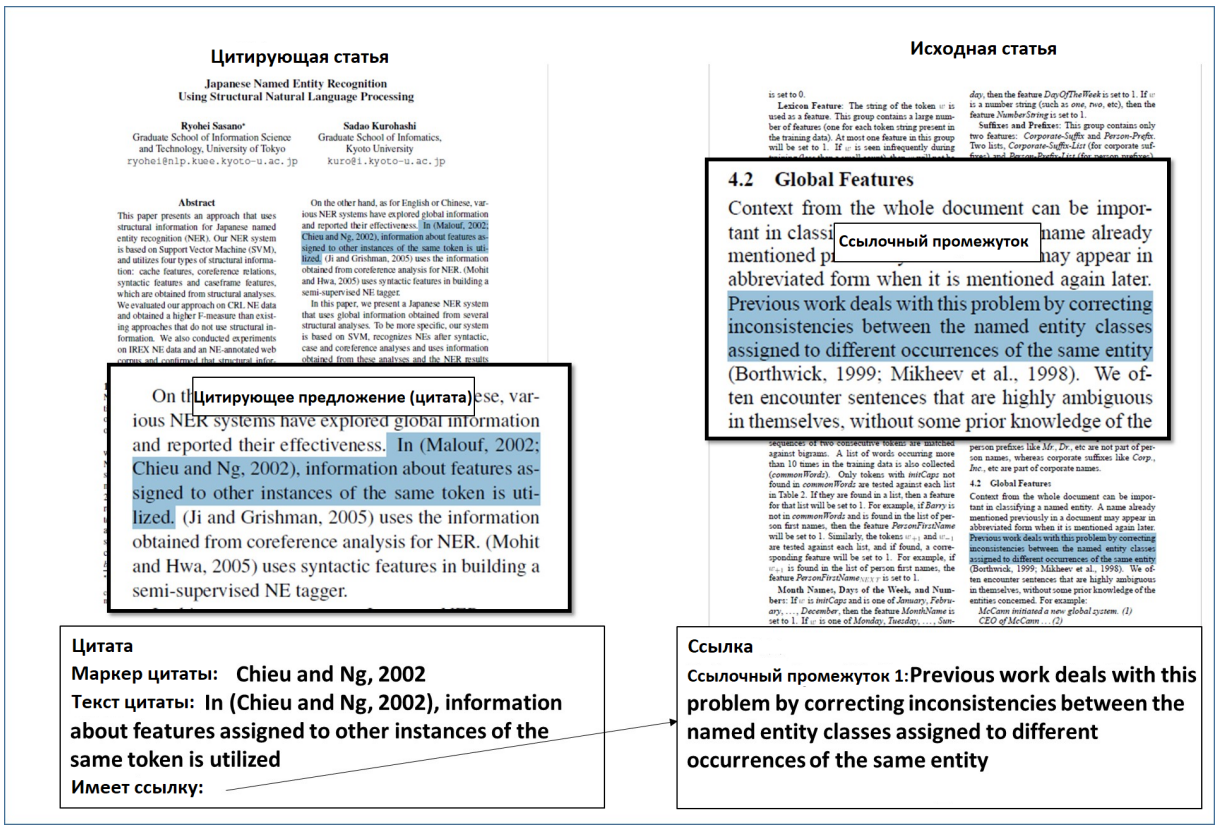


Рис. 10: CL-SciSumm 2018: Пример аннотирования цитируемой и цитирующей статей.

В качестве обучающей выборки предлагается использовать датасет CL-SciSumm 2018 [7] из 40 примеров реферируемых научных статей из области компьютерной лингвистики и их цитирующих статей, а также 20 примеров для тестирования. CL-SciSumm 2018 обладает следующими свойствами:

- К каждой реферируемой статье прилагается не менее 10 цитирующих статей.
- В каждой цитирующей статье выделены цитирующие фрагменты, а также соответствующие им цитируемые (ссылочные) фрагменты в исходной статье. Пример данных из корпуса приведён на Рис.10.
- Для каждой из реферируемых статей предоставляется три вида «эталонных» рефератов:

– Аннотация, написанная авторами статьи (annotation summary);

– Суммаризация на основе предложений цитируемых (ссылочных) фрагментов (community summary);

– Суммаризация на основе полного текста статьи (human summary).

Для обучения модели выделения цитирующего фрагмента из данных CL-SciSumm 2018 был собран набор данных с примерами вида <контекст цитаты, цитирующий фрагмент>. В качестве контекста цитаты были выбраны предложение, непосредственно содержащее цитату, и по 3 предложения справа и слева в рамках параграфа. Таким образом, максимальная длина контекста состоит из 7 предложений. Характеристики выборок представлены в Таб. 2, а распределение длин контекстов на в Таб. 3.

	Train	Test
# примеров в выборке	740	334
# цитируемых документов	40	20
# предложений в контексте	1.21	1.24
# предложений в цитирующем фрагменте	6.6	6.4

Таблица 2: Характеристики выборки для поиска цитирующих фрагментов.

# предложений	Train	Test
1	624	270
2	88	51
3	20	10
4	5	2
5	1	1
6	2	–

Таблица 3: Распределение длин цитирующих контекстов в предложениях.

Качество модели измерялось по метрике F_1 между токенами правильного фрагмента и предсказанного. В результате качество обученной модели лишь на 1% побило

бейзлайн. Однако, можно заметить, что обучаемая модель имеет тенденцию к предсказанию больших фрагментов. Если разбить тестовую выборку по длине фрагмента в токенах на 4 равные части, то модель работает на 3% хуже на коротких примерах, но даёт преимущество в 4 % на длинных фрагментах. Полученные результаты представлены в Таб. 4. Стоит отметить, что дополнительные предложения из контекста могут оказаться полезными в рамках задачи суммаризации. Примеры, где модель работает лучше бейзлайна и ошибается приведены на Рис. 11.

Число токенов	baseline	SciBERT
$(0, Q_1]$	**0.98**	0.95
$(Q_1, Q_2]$	0.94	0.94
$(Q_2, Q_3]$	0.92	0.92
$(Q_3, max\_len]$	0.81	**0.85**
$(0, max\_len]$	0.91	**0.92**

Таблица 4: F_1 между правильным фрагментов и предсказанным на данных CL-Scisumm 2018.

baseline: These are as described in (Mitchell and Lapata, 2008).

SciBERT: *We use the compositionality functions, simple addition and simple multiplication to build compositional vectors Vwr1+wr2 and Vwr1xwr2. These are as described in (Mitchell and Lapata, 2008).*

baseline: *Goldberg and Tsarfaty (2008) propose a generative joint model.*

SciBERT: Goldberg and Tsarfaty (2008) propose a generative joint model. This paper is the first to use a fully discriminative model for joint morphological and syntactic inference on dependency trees.

Рис. 11: Примеры правильной и ошибочной работы модели SciBERT . Цитата выделена красным цветом, а правильный фрагмент – курсивом (baseline).

6.2 Суфлёр, возвращающий цитируемые фразы

Для обучения классификатора на основе данных CL-SciSumm 2018, структура которых описана выше, были составлены обучающая и тестовая выборки из пар (s, c),

где s – предложение статьи, c – предложение из цитирующего фрагмента. Для каждой пары была определена метка: 1, если s входит в соответствующий цитируемый (ссылочный) фрагмент, 0 – иначе. Характеристики выборок представлены в Таб. 5. Из-за сильного дисбаланса классов для обучения классификатора отрицательные примеры сэмплировались в пропорции 4:1 к положительным примерам.

	Train	Test
# примеров в выборке	121583	2687
# отрицательных примеров	120687	26650
# положительных примеров	896	221

Таблица 5: Характеристики выборки для поиска цитируемых фрагментов.

Для составления реферата были последовательно выбраны самые релевантные фразы до достижения 250 слов. В качестве метрики качества работы суфлёра использовалась метрика $ROUGE_2(f1)$ в сравнении с эталонным рефератом. Полученные результаты также сравнивались с качеством моделей, представленных на соревнование в 2020 году [17]. Результаты представлены в Таб. 6. Модель SciBERT показала наилучшие результаты на всех датасетах.

	Community summary	Human summary
baseline	0.19	0.1
MLP	0.33	0.22
boosting	0.30	0.20
SciBERT	**0.36**	**0.23**
best 2020	0.27	0.22

Таблица 6: $ROUGE_2(f_1)$ на CL-SciSumm 2018 суфлёра, возвращающего фразы из статьи, близкие к цитирующим фразам.

6.3 Оценка качества всех суфлёров

Автоматическая оценка качества системы из всех суфлёров проводилась на собранной выборке из обзорных частей, характеристики которой описаны в разделе

«Сбор данных». Генерация реферата производилась согласно алгоритму, описанному в разделе «Оценка качества». В качестве сценария обзора использовалось ранжирование по году публикации. Суфлёры, возвращающие цитирующие цитируемые фразы, были обучены на наборе CL-Scisumm 2018. В качестве экстрактивного суфлёра использовалась модель TextRank.

В результате постобработки были отброшены фразы, состоящие менее, чем из 5 и более, чем из 45 слов. Также личные и притяжательные местоимения заменялись на имена соответствующих авторов статьи.

Была проведена оценка качества для каждого суфлёра по отдельности, а также для всех суфлёров в совокупности. Качество полученных рефератов представлено в Таб. 7. Для собранного датасета лучше всего работает суфлёр, выдающий цитирующие фразы. Это можно объяснить тем, что авторы для описания статьи в разделе «Related Work» зачастую используют цитаты других авторов на эту статью. В совокупности, суфлёры дают прирост по качеству на 7 %. Также, если использовать гиперпараметр $k = 3$, то удаётся поднять качество ещё на 5 %. Таким образом, затратив немного больше усилий на просмотр первых трёх фразы из выдачи суфлёров, пользователь сможет заметно улучшить реферат. На Рис. 12 представлен пример получившегося реферата и «эталонного» раздела «Related Work». Полученные результаты подтверждают гипотезу о том, что вместо автомата пользователю нужна система подсказок, из которой он будет самостоятельно выбирать фразы.

	$top-1$	$top-3$
Abstract	0,11	0,13
Overview	0,08	0,11
Citance	0,16	0,22
Reference	0,13	0,15
All	**0,21**	**0,26**

Таблица 7: $ROUGE_2(f_1)$ системы MAHS на подборке S2ORC.

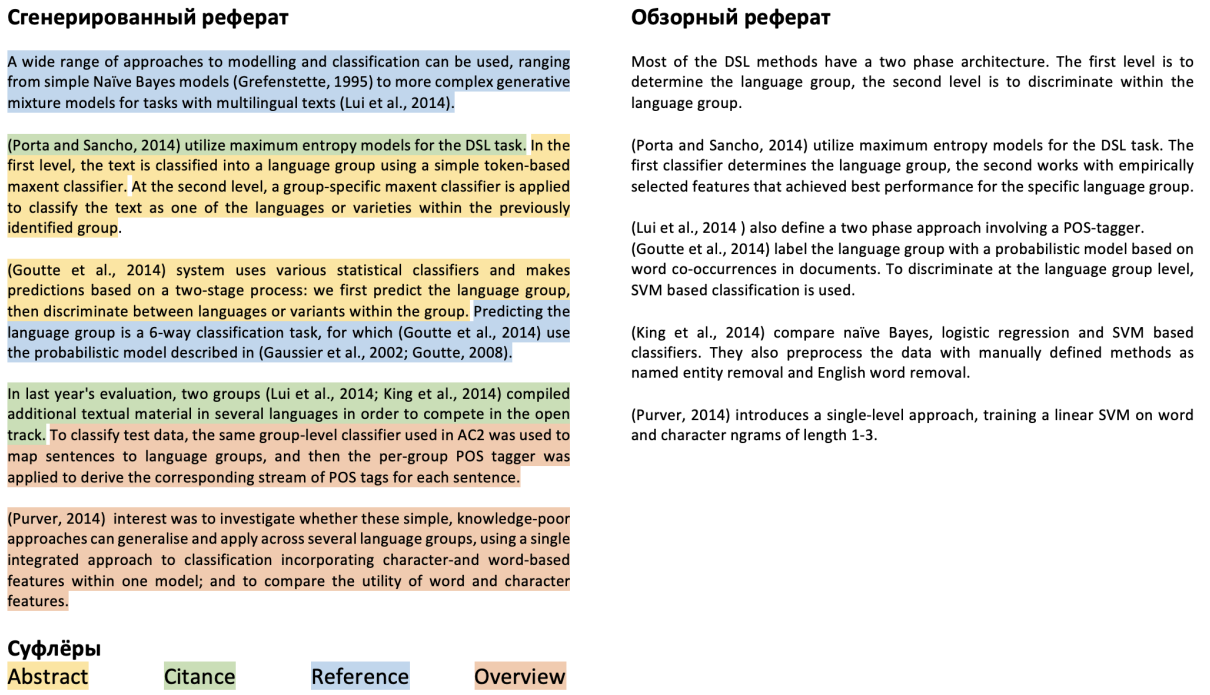


Рис. 12: Пример работы системы MAHS (автоматически сгенерированный реферат, top-3), фразы различных суфлёров выделены цветами.

7 Ресурс «Мастерская знаний»

«Мастерская знаний» – поисково-рекомендательный сервис для формирования и анализа тематических подборок англоязычных научных статей. В рамках сервиса был создан прототип предложенной в работе системы MAHS. Пользовательский интерфейс системы показан на Рис. 13.

Интерфейс позволяет составлять реферат для собранной тематической коллекции, используя фразы-подсказки от суфлёров. В пользовательском интерфейсе суфлёр – это кнопка, нажатие которой перестраивает ранжированный список фраз. При переключении на соответствующий интерфейс, сначала синтезируется сценарий реферата, который при желании может быть скорректирован пользователем. Затем, пользователь составляет реферат в текстовом поле, выбирая нужных суфлёров и используя фразы из предложенного списка. Текстовое поле предоставляет возможность не только использовать готовые фразы, но и редактировать их, а также вводить любой текст.

Стоит отметить, что с помощью логов сервиса можно в дальнейшем производить сбор данных, а также оценивать качество работы системы как количество фраз из выдачи, которые пользователи использовали в итоговом обзоре. Кроме того, можно оценивать качество работы каждого суфлёра как среднюю позицию тех его фраз, которые пользователи отбирали для обзора.

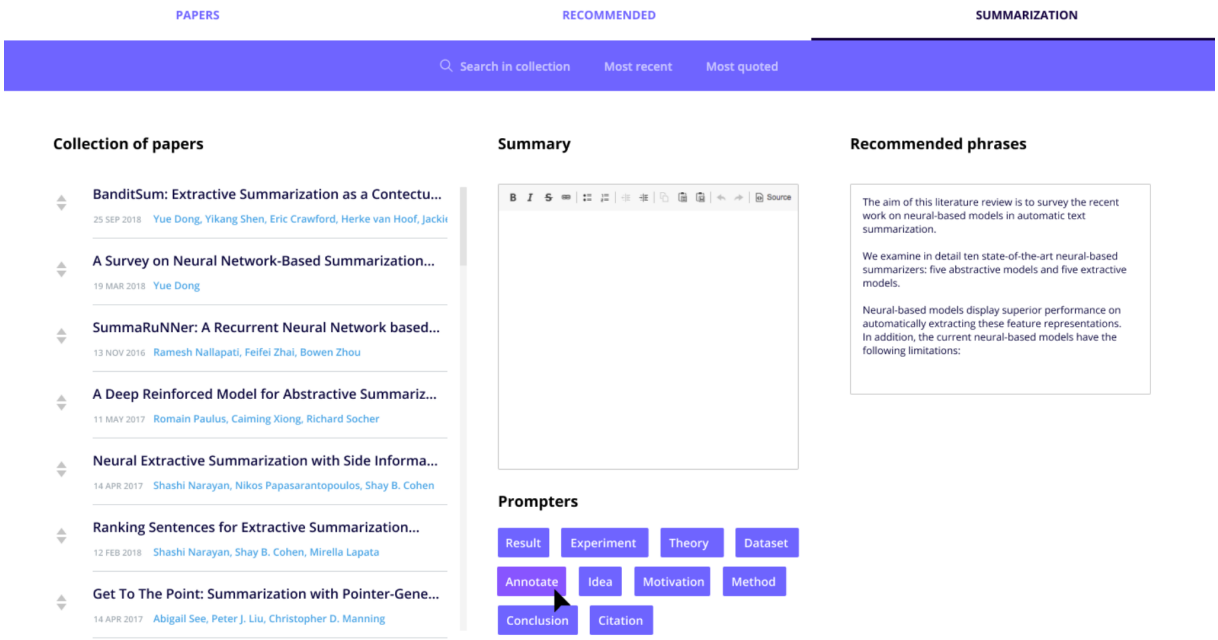


Рис. 13: Интерфейс системы MAHS в сервисе «Мастерская знаний».

8 Заключение

В данной работе предложен новый подход к решению задачи полуавтоматической суммаризации (machine aided human summarization, MAHS) подборок научных публикаций, основанный на ранжировании документов в подборке и построении различных списков фраз-подсказок с помощью нескольких функций ранжирования. Особенность предложенного подхода заключается в том, что пользователь может сам задавать сценарий реферата в зависимости от своих целей, последовательно выбирая фразы для продолжения обзора.

На защиту выносятся следующие результаты:

- Предложена декомпозиция задачи автоматизированной авторской суммаризации научных статей на подзадачи классификации, ранжирования, информационного поиска и автоматической суммаризации.

- Собрана обучающая выборка для задачи суммаризации подборок научных статей на основе обзорных частей.

- Предложен способ для автоматической оценки качества предложенной технологии.

- В экспериментах показано преимущество предложенного подхода по сравнению с автоматическими решениями.

- Создан прототип технологии в поисково-рекомендательном сервисе «Мастерская знаний».

Промежуточные результаты работы были представлены в виде доклада на 20-й Всероссийской конференции с международным участием «Математические методы распознавания образов» [24].

В рамках дальнейших исследований планируется экспертная оценка качества работы системы с помощью сервиса «Мастерская знаний» и развитие системы с помощью новых суфлёров, выделяющих важные фразы относительно аспектов «актуальности», «новизны», «методов», «выводов», «преимуществ», «недостатков» и т.д.

В заключении заметим, что предлагаемый способ полуавтоматической суммаризации может также рассматриваться как способ нелинейного чтения, когда перед пользователем стоит задача не только разобраться в мало знакомой для него области по обширной тематической коллекции, но и одновременно произвести информационный продукт в виде обзора.

Список литературы

[1] Attention is all you need / Ashish Vaswani, Noam M. Shazeer, Niki Parmar et al. // *ArXiv*. — 2017. — Vol. abs/1706.03762.

[2] Bag of tricks for efficient text classification / Armand Joulin, Edouard Grave, Piotr Bojanowski, Tomas Mikolov // EACL. — 2017.

[3] Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension / M. Lewis, Yinhan Liu, Naman Goyal et al. // *ArXiv*. — 2020. — Vol. abs/1910.13461.

[4] *Beltagy Iz, Lo Kyle, Cohan Arman.* Scibert: A pretrained language model for scientific text // EMNLP. — 2019.

[5] Bert: Pre-training of deep bidirectional transformers for language understanding / J. Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova // NAACL-HLT. — 2019.

[6] *Blei David M., Ng A., Jordan Michael I.* Latent dirichlet allocation // *J. Mach. Learn. Res.* — 2003. — Vol. 3. — Pp. 993–1022.

[7] The cl-scisumm shared task 2018: Results and key insights / Kokil Jaidka, Michihiro Yasunga, Muthu Chandrasekaran et al. — 2018.

[8] *Cohan Arman, Goharian Nazli.* Contextualizing citations for scientific summarization using word embeddings and domain knowledge // *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval.* — 2017.

[9] *Conroy John M., Davis S.* Section mixture models for scientific document summarization // *International Journal on Digital Libraries.* — 2017. — Vol. 19. — Pp. 305–322.

[10] Exploring the limits of transfer learning with a unified text-to-text transformer / Colin Raffel, Noam M. Shazeer, Adam Roberts et al. // *ArXiv.* — 2020. — Vol. abs/1910.10683.

[11] *Haghighi A., Vanderwende Lucy.* Exploring content models for multi-document summarization // HLT-NAACL. — 2009.

[12] *Lin Chin-Yew.* Rouge: A package for automatic evaluation of summaries // ACL 2004. — 2004.

[13] *Liu Yang.* Fine-tune bert for extractive summarization // *ArXiv.* — 2019. — Vol. abs/1903.10318.

[14] *Meena Chandra, Shukla A.* Human aided text summarizer "saar "using reinforcement learning. — 2014. — 09.

[15] *Mei Qiaozhu, Zhai ChengXiang.* Generating impact-based summaries for scientific literature // ACL. — 2008.

[16] *Mihalcea Rada, Tarau Paul.* Textrank: Bringing order into text // EMNLP. — 2004.

[17] Overview and insights from the shared tasks at scholarly document processing 2020: Cl-scisumm, laysumm and longsumm / Muthu Kumar Chandrasekaran, Guy Feigenblat, Eduard Hovy et al. // Proceedings of the First Workshop on Scholarly Document Processing. — Association for Computational Linguistics, 2020. — Pp. 214–224.

[18] *Özsoy Makbule Gülçin, Alpaslan F., Çiçekli I.* Text summarization using latent semantic analysis // *Journal of Information Science.* — 2011. — Vol. 37. — Pp. 405 – 417.

[19] Pegasus: Pre-training with extracted gap-sentences for abstractive summarization / Jingqing Zhang, Y. Zhao, Mohammad Saleh, Peter J. Liu // *ArXiv.* — 2020. — Vol. abs/1912.08777.

[20] *Reimers Nils, Gurevych Iryna.* Sentence-bert: Sentence embeddings using siamese bert-networks // EMNLP/IJCNLP. — 2019.

[21] S2orc: The semantic scholar open research corpus / Kyle Lo, Lucy Lu Wang, Mark Neumann et al. // ACL. — 2020.

[22] Scisummnet: A large annotated corpus and content-impact models for scientific paper summarization with citation networks / Michihiro Yasunaga, Jungo Kasai, Rui Zhang et al. // AAAI. — 2019.

[23] *А.В. Власов*. Методы полуавтоматической суммаризации подборок научных статей. — Магистерская диссертация, 2020.

[24] Полуавтоматическая суммаризация тематических подборок научных публикаций: задачи и подходы / Крыжановская С., Власов А., Еремеев М., Воронцов К. — Математические методы распознавания образов: Тезисы докладов 20-й Всероссийской конференции с международным участием, 2021.