САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

КАФЕДРА ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

Илямакова Наталья Юрьевна

Выпускная квалификационная работа бакалавра

Исследование методов машинного обучения для автоматического реферирования документов

Направление 01.03.02

Прикладная математика и информатика

Научный руководитель,
кандидат физ.-мат. наук,
доцент
Добрынин В. Ю.

Санкт-Петербург

2018

Содержание

Введение 3
Постановка задачи 6
Обзор литературы 8
Глава 1. Взаимная информация 11
Глава 2. Кластеризация 13
2.1. K – средних.....13
2.2. Information Bottleneck 14
2.2.1 Алгоритм 15
2.2.2 Дивергенция Кульбака-Лейблера.....15
2.2.3 Дивергенция Дженсена-Шеннона 16
Глава 3. Эксперименты и оценки результатов 17
3.1. Описание данных.....17
3.2. Предобработка данных 17
3.3. Выбор параметра β для Information Bottleneck.....19
3.4. Тесты, направленные на оценку качества кластеризации 21
3.5. Тесты, направлены на проверку работы взаимной информации в качестве меры для вычисления важности слов 22
3.6. Подходы к построению рефератов.....24
3.7. Алгоритм.....27
3.8. Примеры построения рефератов.....29
3.9. Сравнение результатов реферирования.....32

Заключение 36
Список литературы 37

Введение

Под рефератом или аннотацией будем понимать краткое изложение текстового документа, передающее его основное содержание [17]. Выделение из огромных текстов основных мыслей и идей волновало людей со времен возникновения книгопечатания. Но особенно задача автоматического реферирования приобрела актуальность в наши дни, в эпоху развития современных цифровых технологий, когда объемы данных безостановочно растут. В частности, это касается новостных сообщений, когда в многочисленных интернет-источниках ежедневно появляются миллионы новых статей.

Существует несколько классификаций рефератов. По полноте изложения выделяются информативные и индикативные [17]. Информативные рефераты должны сжимать исходный текст, в них полно отражается информация, излагаемая в оригинале, тем самым информативный реферат заменяет исходный текст. Индикативные рефераты более кратки, они должны предоставлять достаточно информации для принятия решения, стоит ли обращаться к первоисточнику, примерами индикативных рефератов являются веб-сниппеты.

Для реферирования новостных сообщений часто применяется многодокументное реферирование [8,17], когда реферат формируется не по одному документу, а по группе документов, посвященных одной теме или описывающих одно событие. В рамках такой задачи из разных новостных сообщений выбираются важные фрагменты и формируется один реферат, полно описывающий событие.

Существуют два основных подхода для решения задачи автоматического реферирования [8]. Первый из них представляет из

себя извлечение из текстов наиболее важных предложений без изменения самих предложений. А второй подход, в отличие от простого извлечения, подразумевает под собой перефразирование основных моментов, базируясь на более глубоком лингвистическом и семантическом анализе. Он лучше сокращает первоначальный текст, но подобные системы намного сложнее разрабатывать, поэтому в ходе исследований был рассмотрен первый вид автоматического реферирования, основанный на выделении самых важных и информативных частей текста.

Таким образом, все предложения одного документа делятся на два класса: на важные и информативные, которые войдут в реферат, и на те, чья важность незначительна. Такая задача решается с помощью бинарного классификатора. Для реализации необходима обучающая выборка с уже размеченными предложениями (метка «1», если предложение подходит для реферата, метка «0» — если не подходит). В виду отсутствия обучающей выборки и того, что на практике довольно затруднительно получить уже размеченные данные для русскоязычных статей, данная концепция не подошла для решения задачи автоматического реферирования в рамках данной исследовательской работы.

В основу работы легла идея о том, что важность предложения напрямую зависит от важности входящих в него слов [9]. И прежде чем составлять реферат, необходимо выделить наиболее важные, полезные слова. В качестве меры полезности слова нередко рассматривают частотные характеристики слов – частота встречаемости слова и обратная документная частота (TF-IDF) [3]. Но в ходе данной работы полезность слов определялась с помощью такой меры как взаимная информация, которой посвящена Глава 1.

Если рассмотреть две новостные статьи, принадлежащие разным темам, то и важные слова для статей будут различаться, так как каждая тематически ориентированная группа документов имеет специализированные термины, которые в документах, относящихся к другой группе, встречаются крайне редко или вовсе не встречаются. Для выявления специализированных терминов необходимо сначала разбить все документы на более узкие тематические группы – кластеры, а затем, используя взаимную информацию между кластером и словом в качестве меры полезности слова, извлекать наиболее важные слова уже не для всей коллекции, а для группы документов, объединенных в один кластер. Методам кластеризации посвящена Глава 2, а сравнительный анализ результатов их работы на имеющихся данных представлен в Главе 3.

Постановка задачи

Целью данной работы является создание алгоритма для построения рефератов к новостным статьям, включающих в себя самые важные предложения.

Для достижения данной цели были поставлены следующие подзадачи:

1. Провести предварительную обработку текстов.
2. Реализовать метод кластеризации Information Bottleneck.
3. Кластеризовать документы, используя реализованный Information Bottleneck и готовый метод к-средних из python библиотеки scikit-learn [15]. Сравнить результаты кластеризации.
4. Проверить работу взаимной информации между словами и кластерами в качестве меры важности слов для построения аннотаций.
5. Построить рефераты для новостных сообщений, используя меру полезности слов и данные, полученные в результате кластеризации.
6. Сравнить полученные рефераты с экспертной оценкой и с рефератами, построенными другими системами.

Формализация задачи

Имеется коллекция из N документов $X = (x_1, x_2, \dots, x_n)$.

Под документом будем понимать текст новостной статьи.

Каждый документ разбивается на предложения s_{ij}, где s_{ij} – j-ое предложение в i–ом документе. $S_i = (s_{i1}, s_{i2}, \dots, s_{im})$, где m – количество предложений в i–ом документе.

W_{ij} – вес предложения s_{ij} , который вычисляется по следующей формуле

$$W_{ij} = \sum_{k=1}^{K_{ij}} w_k ,$$

где K_{ij} – количество слов в j-ом предложении i-ого документа,

w_k – вес k-ого слова,

$$w_k = A(word_k),$$

где $A(word)$ – функция, характеризующая меру полезности слова. Необходимо для $\forall i$ получить

$$R_i = (r_{i1}, r_{i2}, \dots, r_{iL}),\ R_i \subset S_i ,$$

где L – желаемый размер реферата, а элементы r_{il} для $l = \overline{1, L}$ – это предложения с наибольшим значением веса.

Обзор литературы

В самой первой работе, посвященной автоматическому реферированию текста [9], высказывались идеи о том, что важность участка текста (в рамках одного документа) высчитывается исходя из важности входящих в него слов. Причем необходимо исключить общие слова и наиболее редкие. Также предполагается, что слова, расположенные близко, должны друг друга усиливать, тем самым увеличивая важность фрагмента текста, к которому они принадлежат. В еще одной ранней работе [6] внимание уделялось не только одному компоненту значимости предложения (наличию высокочастотных слов), описанные в ней методы ранжируют предложения в зависимости от его расположения в корпусе документа и наличия в предложении слов из заголовка.

Работа [5] дает хорошее представление о последних тенденциях и достижениях в области автоматического реферирования. В ней представлен прекрасный обзор наиболее используемых методов для реферирования путем выделения наиболее важных предложений в тексте: методы, базирующие на частотных характеристиках слов – частоте слов и обратной документной частоте (TF-IDF); подходы, основанные на тематическом моделировании (Latent Semantic Analysis) и вероятностном тематическом моделировании (Latent Dirichlet Allocation); методы на основе графов (PageRank) и машинного обучения. Кроме того, автор уделил внимание методам оценки готовых рефератов. Была выделена наиболее широко используемая метрика для автоматической оценки — ROUGE и экспертное оценивание, например, The Document Understanding Conference (DUC) и The Text Analysis Conference (TAC).

Также для решения задачи автоматического реферирования широко применяют методы машинного обучения. Например, в работах [1, 11] использовали один из методов машинного обучения с учителем. Обученный классификатор, который на основе признаков предложения (длины, расположения в документе, наличия именованных сущностей, значений TF-IDF и т. д.) определял, какие предложения должны войти в реферат. В работе [1] предложения классифицировались с помощью метода опорных векторов (SVM), в [11] с помощью наивного байесовского классификатора, в обоих случаях результат работы классификатора примерно на половину совпадал с оригинальным квазирефератом.

Необычный подход продемонстрирован в работе [10], где важные предложения выделяются с помощью оптимизации методом биогеографии (Biogeography-based optimization method). В качестве основы была рассмотрена простейшая математическая модель миграции популяции по местообитаниям. Вместо максимального числа сред обитания было принято общее число предложений в документе, а в качестве входного вектора параметров, определяющих среду обитания — матрица схожести, посчитанная между предложениями и вектором важных для документа слов. На выходе вместо списка мест обитания, которые будут решениями задачи оптимизации, получим набор предложений, подходящих для реферата.

Для построения аннотаций активно используют нейронные сети. Несмотря на то, что рекуррентные нейронные сети показывают высокую эффективность, их сложно реализовать на практике в связи с тем, что такие модели требуют больших вычислений и не все машины обладают соответствующей вычислительной способностью. В основе подхода, описанного в работе [13], лежит использование нейронных сетей

прямого распространения (Feedforward). Модель обучается и оценивается на стандартном наборе данных DUC 2002. Построенная нейронная сеть состояла из одного входного слоя, одного скрытого и одного выходного слоя. На вход подавались предложения документа, представленные в векторной форме. Сначала с помощью модели word2vect слова предложений представлялись в виде векторов, а затем создавалось векторное представление самих предложения. Полученная модель способна составлять рефераты произвольного размера, создавая реферат фиксированного размера и затем рекурсивно подавая их обратно в сеть до получения желаемого результата.

Глава 1. Взаимная информация

В качестве меры $A(word)$, характеризующую полезность слова $word$ рассмотрена такая мера как взаимная информация [2].

Взаимная информация $I(X,Y)$ – функция двух случайных величин X и Y , измеряющая количество информации, которое содержится в одной случайной величине относительно другой.

Для дискретного случая взаимная информация двух случайных величин X, Y определяется как

$$I(X,Y) = \sum_{x\in\mathbb{X},y\in\mathbb{Y}} p(X=x,Y=y)\log\frac{p(X=x,Y=y)}{p(X=x)p(Y=y)}$$

В [3] рассматривают применение взаимной информации при выборе признаков для решения задачи классификации. Из всего множества слов для каждого класса отбирается такое подмножество слов, чьи значения полезности являются наибольшим, где в роли меры полезности выступает $I(W,\mathrm{C})$ – взаимная информация между классом c и словом w, где $\mathrm{c}\in\mathbb{C}$, $w\in\mathbb{W}$, $\mathbb{W}$ – множество слов, $\mathbb{C}$ – множество классов.

В связи с тем, что большая часть полученных текстовых документов, для которых ставилась задача построения рефератов, не имела меток о принадлежности к той или иной тематике и что на практике чаще встречаются данные для которых обучающая выборка отсутствует, в ходе исследовательской работы вместо классов использовались кластеры.

Пусть множество документов $\mathbb{X}$ разбили на фиксированное число кластеров. Теперь необходимо вычислить меру полезности каждого слова $w\in\mathbb{W}$, для каждого кластера $t\in\mathbb{T}$, где $\mathbb{T}$– множество кластеров.

Если w – это слово, t – кластер, то $I(W,T)$ – величина, которая характеризует количество информации о принадлежности документа x к кластеру t в зависимости от того, присутствует ли в нем слово w или нет. Здесь W – случайная величина, принимающая значение $u_w = 1$, если документ x содержит слово w, $u_w = 0$, если документ x не содержит слово w, а T – случайная величина, принимающая значение $u_t = 1$, если документ x принадлежит кластеру t, $u_t = 0$, если документ x не принадлежит кластеру t.

Тогда согласно [3] и с учетом замены классов на кластеры имеем

$$I(W,T) = \sum_{u_w \in \{0,1\}, u_t = \{0,1\}} p(W = u_w, T = u_t) \log \frac{p(W = u_w, T = u_t)}{p(W = u_w)p(T = u_t)}$$

И после использования оценок максимального правдоподобия итоговая формула имеет следующий вид

$$I(W,T) = \frac{N_{11}}{N}\log_2 \frac{NN_{11}}{N_1 N_1} + \frac{N_{01}}{N}\log_2 \frac{NN_{01}}{N_0 N_1} + \\ + \frac{N_{10}}{N}\log_2 \frac{NN_{10}}{N_1 N_0} + \frac{N_{00}}{N}\log_2 \frac{NN_{00}}{N_0 N_0}, \qquad (1)$$

где N – общее число документов,

N_{11} – количество документов содержащих w, принадлежащих t,

N_{01} – количество документов не содержащих w, принадлежащих t,

N_{10} – количество документов содержащих w, не принадлежащих t,

N_{00} – количество документов не содержащих w, не принадлежащих t,

$N_1 = N_{10} + N_{11}$ – общее количество документов, содержащих w,

$N_0 = N_{01} + N_{00}$ – общее количество документов, не содержащих w.

Глава 2. Кластеризация

Под кластеризацией понимают разбиение множества входных данных на подмножества, именуемые кластерами, причем необходимо получить такое разбиение, чтобы кластеры получились однородными внутри и максимально отличались от остальных кластеров.

2.1. K – средних

K-средних [7] – наиболее распространенный и часто используемый метод кластеризации. Его задача заключается в разбиении множества элементов $X = (x_1, x_2, \dots, x_N)$ на заранее известное число кластеров $T_k = \{x_j\},$ где $k = \overline{1,\ K}$, K – число кластеров, а $j \in J_k$, где J_k– множество индексов элементов, попавших в кластер k.

Цель алгоритма – минимизировать среднеквадратичное отклонение элементов кластеров от центров кластеров , к которым эти элементы принадлежат

$$\sum_{k=1}^{K} \sum_{x \in T_k} |x - \mu_k|^2,$$

где $\mu_1, \mu_2, \dots, \mu_K$ – центры кластеров.

Метод заключается в том, что на каждой итерации перевычисляется центр μ_k для каждого кластера, полученного на предыдущем шаге, а затем элементы заново разбиваются на кластеры в зависимости от того, какой из новых центров оказался ближе. Критерий остановки алгоритма – отсутствие изменений в перераспределении кластеров на новой итерации.

2.2. Information Bottleneck

Information Bottleneck [14, 16] – метод для мягкой кластеризации, результат его работы – не конкретный номер кластера, к которому принадлежит документ x, а вероятность попадания документа x в кластер t. Пусть известно совместное распределение $p(x,w)$, $x \in \mathbb{X}$, $w \in \mathbb{W}$, искомая вероятность $p(t|x)$, $x \in \mathbb{X}$, $t \in \mathbb{T}$, будет находиться путем минимизации функционала $L(p(t|x))$

$$L\big(p(t|x)\big) = I(X,T) - \beta I(T,W) \xrightarrow[p(t|x)]{} min \qquad (2)$$

$I(T,X)$ характеризует сжатие данных, $I(T,W)$ – релевантность, главная цель – найти компромисс между сжатием данных и сохранение релевантности, за это отвечает параметр β, который подбирается экспериментально.

Основную идею этого метода иллюстрирует Рисунок 1.

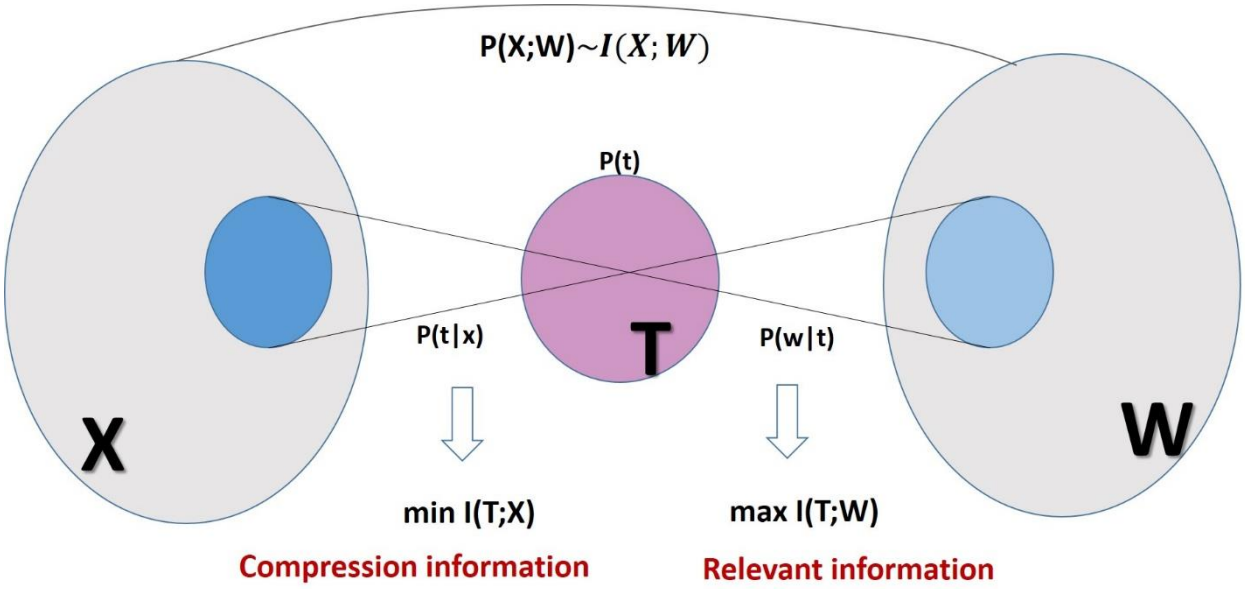

Рисунок 1. Информация между X и W сжимается через компактное представление T

Теорема [16]. Оптимальное решение имеет вид

$$p(t|x) = \frac{p(t)}{Z(x,\beta)} e^{-\beta D_{KL}(p(w|x)||p(w|t))}, \quad \forall t \in \mathbb{T}, \forall w \in \mathbb{W} \quad (3)$$

где $Z(x,\beta)$ *– нормализующая функция,* $D_{KL}(p(w|x)||p(w|t))$ *– дивергенция Кульбака –Лейблера (см. 2.2.2).*

2.2.1. Алгоритм

Входные данные:

- *Совместное распределение* $p(x,w)$
- β
- ε *– желаемая точность*
- M *– количество кластеров*

Выходные данные:

- $p(t|x)$ *– распределение документов по кластерам.*

Инициализация:

- $p(t|x)$ *– случайная инициализация, при условии* $\sum_{i=1}^{M} p(t_i|x) = 1, \quad 0 \le p(t_i|x) \le 1$ *для* $\forall x \in \mathbb{X}$

Цикл пока верно:

- $p_{m+1}(t) = \sum_{x\in\mathbb{X}} p(x) p_m(t|x), \quad \forall t \in \mathbb{T},$
- $p_{m+1}(w|t) = \frac{1}{p_{m+1}(t)} \sum_{x\in\mathbb{X}} p_{m+1}(t|x)\, p(x,w), \quad \forall t \in \mathbb{T}, \forall w \in \mathbb{W}$
- $p_{m+1}(t|x) = \frac{p_m(t)}{Z_{m+1}(x,\beta)} e^{-\beta D_{KL}[p(w|x)p(w|t)]}, \quad \forall t \in \mathbb{T}, \forall x \in \mathbb{X},$

если $JS_{\frac{1}{2},\frac{1}{2}}[p_{m+1}(t|x)||p_m(t|x)] \le \varepsilon, \ \forall x \in \mathbb{X}$, *(см. 2.2.3)*

закончить.

2.2.2. Дивергенция Кульбака-Лейблера

Дивергенция Кульбака-Лейблера $D_{KL}(P||Q)$ – величина, которая характеризует удаленность вероятностного распределения P от другого

вероятностного распределения Q. Для двух дискретных распределений P и Q с элементарными событиями $p_1, \dots, p_n$ и $q_1, \dots, q_n$ соответственно дивергенция Кульбака-Лейблера вычисляется

$$D_{KL}(P||Q) = \sum_{i=1}^{n} p_i \log \frac{p_i}{q_i}$$

и обладает свойствами:

1) $D_{KL}(P||Q) \geq 0$ – неотрицательно
2) $D_{KL}(P||Q) = 0, \text{если } P = Q$
3) $D_{KL}(P||Q) \neq D_{KL}(Q||P)$ – не симметрично

Также имеется связь между взаимной информацией $I(X, W)$ и дивергенцией Кульбака-Лейблера

$$I(X, W) = D_{KL}(\, p(x, w)||p(x), p(w))$$

2.2.3. Дивергенция Дженсена-Шеннона

Дивергенция Дженсена-Шеннона $JS_{\pi_1,\pi_2}(P||Q)$ так же, как и дивергенция Кульбака-Лейблера, является величиной, характеризующей насколько одно вероятностное распределение похоже на другое.

$$JS_{\pi_1,\pi_2}(P||Q) = \pi_1 D_{KL}(P||M) + \pi_2 D_{KL}(M||Q),$$

где $M = \frac{P+Q}{2}$, $0 < \pi_1, \pi_2 < 1$, $\pi_1 + \pi_2 = 1$.

В качестве критерия остановки алгоритма Information Bottleneck, используется дивергенция Дженсена-Шеннона с $\pi_1 = \pi_2 = \frac{1}{2}$

$$JS_{\frac{1}{2},\frac{1}{2}}(P||Q) = \frac{1}{2} D_{KL}(P||M) + \frac{1}{2} D_{KL}(Q||M)$$

Глава 3. Эксперименты и оценки результатов

3.1. Описание данных

Тестирования проводились на двух разных коллекциях документов.

Первая, назовем ее К1, состояла из новостных сообщений из новостного интернет-источника https://lenta.ru/. Основные характеристики коллекции К1: размер коллекции ~30 тыс. документов, средний размер текстов от 8-20 предложений; приблизительно равное количество документов, принадлежащих разным тематикам («Спорт», «Культура», «Наука и техника», «Мир», «Россия», «Бывший СССР»).

Вторая коллекция К2 была получена от компании Digital Design. Основные характеристики документов: размер коллекции ~40 тыс. документов; размеры текстов сильно отличаются (от 10 – 100 предложений); большая часть текстов принадлежала тематикам «Политика», «Общество», «Происшествия».

3.2. Предобработка данных

Для повышения качества результатов необходимо уделить должное внимание первоначальной обработке текстовых документов. Предобработка данных состоит из нескольких этапов.

1) Токенезация

Под этим этапом понимают разбиение текста на более мелкие токены (слова, числа). Слова выделялись с применением регулярных выражений с помощью модуля re [18] (python).

2) Удаление стоп-слов

Стоп-слова — это лексические единицы текста, которые сами по себе не несут смысловой нагрузки и поэтому в целях уменьшения списка уникальных слов для нашей коллекции документов стоп-слова игнорировались. Список стоп-слов для русского языка импортировался из python библиотеки NLTK [12], кроме того этот список был расширен.

3) Лемматизация

Для того, чтобы наш список уникальных слов не пополнялся разными словоформами одного и того же слова, была применена лемматизация — преобразование слов в их словарную форму. Для этого использовался морфологический анализатор pymorphy2 [4].

Также все слова приводились к нижнему регистру, чтобы наш список уникальных слов не рос за счет копирования слов, встречающихся в тексте и с заглавной, и с строчной буквы.

4) Выделения предложений

Так как основной задачей этой работы является извлечение наиболее важных участков текста, необходимо качественно разбивать текст на предложения. В случаях, когда тексты хорошо сформулированы, с задачей хорошо справляются простейшие регулярные выражения. Сложности возникают с обработкой аббревиатур. В целях избежания подобных коллизий, для разбиения предложения использовалась уже готовая функция sent_tokenize() из библиотеки NLTK.

3.3. Выбор параметра β для Information Bottleneck

Качество кластеризации методом IB зависит от значения параметра β из формулы оптимального решения (3), этот параметр регулирует соотношение между сжатием данных и сохранением релевантности.

В качестве критерия точности возьмем такой показатель оценивания кластеризации, как чистота (purity) [19]. Изначально, для тестирования бралась выборка из тех документов, которые имели метки, характеризующие их принадлежность к той или иной тематике. После проведенной кластеризации происходил подсчет количества документов одинаковой тематики, попавших в один кластер, и каждому кластеру присваивалась та тема, с которой у данного кластера нашлось больше всего совпадений. Чистота будет представлять отношение количества верно присвоенных документов к общему числу документов.

$$\text{чистота} = \sum_{k=1}^{K} \frac{1}{N} \times \max_{j}(T_k \cap C_j) \quad , \qquad j = \overline{1,J} \tag{4}$$

где $\{T_k\}_{k=1}^{K}$ – множество кластеров, $\{C_j\}_{j=1}^{J}$ – множество тем, N– общее число документов.

Так как документы коллекций К1 и К2 отличаются друг от друга длинами и тематической ориентированностью, параметр β подбирался отдельно для каждой коллекции.

Для тестирования из каждой коллекции была взята выборка из 900 документов, принадлежащих трем тематикам. Для К1 – «Спорт», «Культура», «Наука и техника», для К2 – «Наука», «Культура», «Экономика», $\varepsilon = 1e - 04$.

Согласно результатам, представленным в Таблице 1, документы

из К1 лучше кластеризуются, когда $\beta = 1.8$, а документы из К2 – когда $\beta = 2.4$. В дальнейших исследовательских экспериментах при вычислении оптимального решения $p(t|x)$ использовались именно эти значения параметра β.

Таблица 1. Чистота кластеризации в зависимости от значения β, %

β	К1	К2
40.0	40%	39%
10.0	50%	50%
5.0	54%	78%
3.1	91%	93%
2.7	93%	95%
2.4	94%	96%
2.1	95%	95%
1.8	97%	68%
1.5	62%	39%
0.6	33%*	33%*

(*) – почти вся коллекция документов оказалась в одном кластере.

Также ниже представлены графики, иллюстрирующие значения функционала $L(p(t|x))$ (4) на каждой итерации.

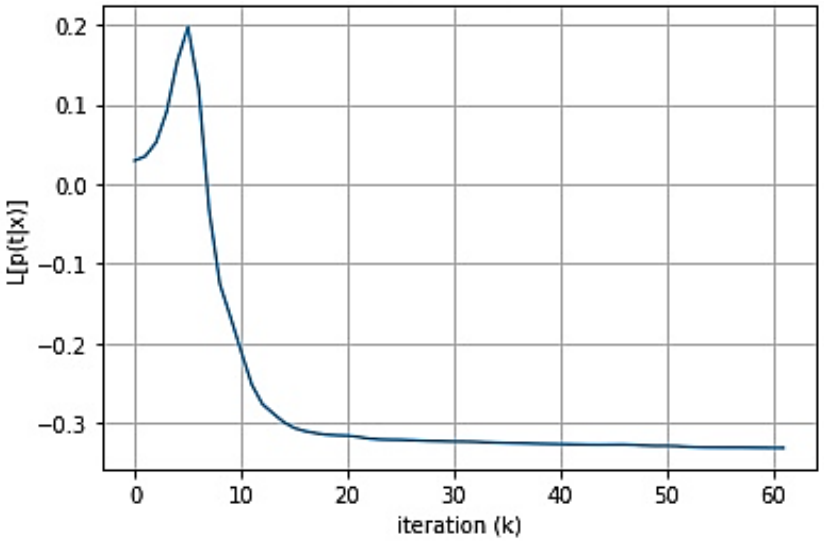

Рисунок 2. Изменения значений функционала L в процессе кластеризации документов из К1

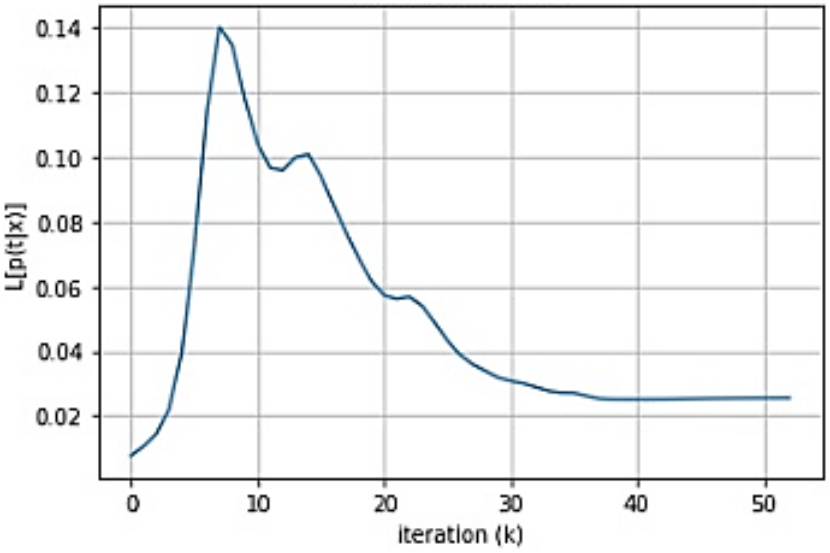

Рисунок 3. Изменения значений функционала L в процессе кластеризации документов из К2

3.4. Тесты, направленные на оценку качества кластеризации

Конечный результат реферирования напрямую зависит от качества кластеров, поэтому необходимо выбрать метод, который лучше кластеризует имеющиеся данные.

Из каждой из двух коллекции $K1$ и $K2$ бралась выборка из N документов, затем проводилась кластеризация одним из двух методов:

K-средних и Information Bottleneck.

Критерием оценивания, как в пункте 3.3, была чистота (4).

Таблица 2. Результаты кластеризации для K1,%

N / Метод	300	600	900
IB	67%	96%	97%
K- средних	68%	84%	84%

Таблица 3. Результаты кластеризации для K2,%

N / Метод	300	600	900
IB	76%	83%	96%
K- средних	75%	73%	73%

В Таблице 2 и Таблице 3 можно заметить, что на имеющихся данных Information Bottleneck работает лучше, поэтому в дальнейшем будет использоваться именно он.

3.5. Тесты, направлены на проверку работы взаимной информации в качестве меры для вычисления важности слов

Для тестирования взята выборка размером в 1200 новостных документов из коллекция $K1$ и 900 документов из К2. Обе выборки

разбивалась на три кластера методом IB. Затем, используя формулу (3) для вычисления взаимной информации $I(U,W)$ выделялись наиболее информативные слова для каждого кластера. В Таблице 4 и Таблице 5 представлены слова с наибольшими значения взаимной информации относительно фиксированного кластера.

Таблица 4. Десять важных слов для кластеров, K1

Кластер №1	**Кластер №2**	**Кластер №3**
Яшин	Импрессионизм	Биопечать
Велоспорт	Иллюстрировать	Паразитизм
Товарищеский	Кончаловский	Кардиолог
Эстафетный	Сикстинский	Нематематический
Керлинг	Кубрик	Баренцев
Шестикратный	Мосфильм	Диффузный
Атлетико	Брандо	Нанофотонник
Мундиаль	Авангардный	Гидрат
Bosco	Симфонический	Тетрациклин
Квят	Кинокомпания	Диагностика

Анализируя результаты, представленные в Таблице 4, можно заметить, что слова кластера №1 объединяет тематика спорт, слова кластера №2 – тематика культура, слова кластера №3 – наука.

Таблице 5 иллюстрирует, что в кластере №1 преимущественно собрались документы, объединенный темой «Культура», в кластере №3 – темой «Наука», для кластера №2 сложнее однозначно определить общую тему, но интуитивно ближе кажется тема «Экономика».

Практические результаты подтвердили, что взаимная информация является хорошим инструментом для выявления важных слов, поэтому

взаимная информация будет применяться в задаче автоматического реферирования.

Таблица 5. Десять важных слов для кластеров, K2

Кластер №1	Кластер №2	Кластер №3
Киплинг	Кибератака	Дерматология
Джоконда	Полковник	Битный
Фадеев	Деноминация	Метеороид
Миланский	Росконтроль	Импланитровать
Земфира	Газпромбанк	Радиотелескоп
Саундрек	Усманов	Позвоночник
Именитый	Жилплощадь	Механика
Онегин	Глобализация	Окислитель
Поленов	Нотариус	Физиологический
Аниматор	Лизинг	Энерготехнология

3.6. Подходы к построению рефератов

1. В реферат R_i новостного сообщения i должны входить наиболее важные предложения, то есть те, чьи веса окажутся наибольшими, а вес предложения пропорционален сумме весов входящих в него слов, $W_{ij} = \sum_{k=1}^{K_{ij}} w_k$, где K_{ij} – количество слов в j-ом предложении i-ом новостном сообщении, w_k – вес k-ого слова. В ходе работы встретилась проблема – предложения с большим количеством слов набирали общий вес за счет своей длины. В целях устранения таких недочетов для подсчета веса предложения учитывались веса только Q наиболее важных слов.

2. Так как в качестве меры полезности слова (или веса слова) рассматривается величина взаимной информации между словом и кластером, то мы имеем набор $MI = (MI_1, \dots, MI_K)$, где для $k = \overline{1, K}$ вектора MI_k имеют разные значения важности слов в зависимости от того, относительно какого кластера считалась взаимная информация. Учитывая, что IB — это метод мягкой кластеризации, есть возможность строить рефераты на основе распределения документа по кластерам. Выбирать важные предложения в зависимости от того, чему равна вероятность принадлежности документа x кластеру t. Рассмотрим два подхода. Первый назовем однокласторным, второй – многокластерным.

a. В однокласторном подходе реферат полностью включит в себя только те предложения, которые будут считаться важными относительно одного кластера, к которому выбранный документ относится с наибольшей вероятностью.
b. В многокластрном будет полностью учитываться распределение документа по кластерам $p(t|x)$. Например, пусть $K = 3$, $p(t|x) = (0.2; 0.1; 0.7)$. Для данного документа реферат на 20% будет состоять из предложений, которые являются важными относительно кластера №1, на 10% — относительно кластера №2, на 70% — относительно кластера №3.

3. Также, очень длинные документы могут включать в себя несколько тематических частей, поэтому имеет следующий подход к построению реферата:

a. Предварительно разбить большие тексты на сегменты одинаковой длины (например, по десять предложений). Получится новый набор из большего числа документов, но

меньших по объему.

b. Затем, кластеризуя новый набор, получить наиболее вероятную тему для каждого сегмента целого документа и рассчитать значения важности слов.

c. Построить реферат для целого документа, включающий в себя важные предложения отдельных сегментов текста.

4. Немаловажным является выбор размера реферата. К этому вопросу можно подступиться с нескольких сторон:

a. Выбрать фиксированное число предложений для реферата.
b. Объем реферата — это доля от объема изначального текста.
c. Зависимость объема реферата от значений важности предложений. Идея заключается в том, что необходимо добавлять предложения в реферат до тех пор, пока разница следующего W_{i+1} значения между предыдущим W_i в отсортированном списке весов не будет больше определенного порога ε, $W_{i+1} - Wi \leq \varepsilon$. Либо пока разница между предложением с наибольшим весом W_1 и предложением W_l, которое мы хотим включить в реферат меньше определенного порога δ, $W_1 - W_l \leq \delta$, $l = \overline{1, N}$.

   Если $\varepsilon = 2, \delta = \frac{W_1}{2} = 5.6$, отсортированный список весов $[11.2, 10.4, 10.2, 9.1, 6.9, 6.1, 5.1, 4.5]$,то в реферат попадут предложения, которым соответствуют первые четыре значения из списка, так как разница между четвертым и пятым элементом превосходит порог ($9.1 - 6.9 > \varepsilon$), $(11.2 - 6.9 < \delta)$.

Таким образом, можно строить рефераты, комбинируя указанные выше способы извлечения важных предложения.

3.7. Алгоритм

1. Для выборки из N документов $X = (x_1, x_2, \dots, x_N)$ создать словарь уникальных слов.
2. Используя метод Information Bottleneck получить распределение данной выборки документов по K кластерам.
3. Отнести документы x_n, $n = \overline{1,N}$ в тот кластер, вероятность попадания в который наибольшая согласно распределению $p(t|x)$, полученному на шаге 2.
4. Для каждого сформировавшегося кластера и для каждого уникального слова из словаря посчитать значение взаимной информации между кластером и словом согласно (1), получим $MI = (MI_1, \dots, MI_K)$, $k = \overline{1,K}$, где $MI_k = (mi_1, \dots, mi_N)$ – вектор значений взаимной информации для кластера k и слова n, $n = \overline{1,N}$.
5. Выбрать документ x_{choice}, для которого будет строиться реферат.
6. Определить значение параметра Q, определяющего количество слов, которые будут учитываться при подсчете весов предложений (3.6. пункт 1), согласно формуле:
$$Q = \frac{2}{3} \times \frac{\text{общее количество слов в тексте}}{\text{количество предложений}}$$
7. Документ x_{choice} разбить на предложения s_j, где s_j – j-ое предложение в выбранном документе. $S_{choice} = (s_1, s_2, \dots, s_J)$, где J – количество предложений в x_{choice}.

8а. Если выбран однокластерный подход (3.6.2. пункт а):

1. Пусть $cluster$ – кластер, для которого значение $p(cluster|x_{choice})$ наибольшее.
2. Для каждого предложения $s_j \in S_{choice}$ посчитать его вес

$Weight_j$

$$Weight_j = \sum_{w \in W_j^{cluster}} w,$$

где $W_j^{cluster}$ – Q наибольших значений $mi_i \in MI_{cluster}$, а i – индексы только тех слов, которые присутствуют в предложении s_j.

3. Определить объема реферата V (3.6. пункт 4).

4. Для реферата из S_{choice} выбрать V предложений с наибольшими $Weight_j$, получим $R_{choice} = (r_1, r_2, \dots, r_V)$.

8b. Если выбран многокластерный подход (3.6.2. пункт b):

1. Для каждого $cluster_k$, $k = \overline{1, K}$
   - Пункт 2 из 7a при $cluster = cluster_k$
   - Определить объема реферата V согласно (3.6.2. пункт a или 3.6.2. пункт b), затем получаем итоговый объем реферата относительно кластера k

     $$V_k = V \times p(cluster_k | x_{choice}) \xrightarrow{\text{округлить до целых}} V_k$$

   - R_k – это V_k предложений с наибольшими $Weight_j$
2. $R_{choice} = R_1 \cup R_2 \cup \dots \cup R_k$ для $k = \overline{1, K}$.

9. R_{choice}

Для случая 3.6. пункта 3 алгоритм слегка модифицируется.

1. Все документы из выборки N документов $X = (x_1, x_2, \dots, x_N)$ разбить на части равных размеров (фиксированное число предложений). Получим новое множество документов

размером S – $X_{new} = (x_1^1, \dots, x_i^n, \dots x_S^N)$, где x_i^n – часть документа x_n.

2. Для $X_{new} = (x_1^1, \dots, x_i^n, \dots x_S^N)$ применить шаги 1-8 алгоритма, описанного выше.
3. $R_{choice} = R_1 \cup R_2 \cup \dots \cup R_{parts}$, где R_i – рефераты к каждой части документа, $i = \overline{1, parts}$, $parts$ – количество частей, на которое разбился x_{choice}.

Реализация алгоритма на языке python представлена в репозитории GitHub [20].

3.8. Примеры построения рефератов

x_1 ={

- Роналду решили продать в китайский клуб за 300 млн евро.
- Как сообщает ТАСС со ссылкой на испанскую прессу, китайцы готовы выложить за форварда 300 миллионов евро.
- Руководство «Реала» перед таким предложением не устояло.
- Примечательно, что испанский клуб продлил контракт с португальцем только в ноябре прошлого года.
- Договор рассчитан до 2021 года.
- Если сделка по продаже все же состоится, Роналду перейдет в китайский клуб в сезоне 2018/19.
- Роналду выступает за «Реал» с 2009 года.
- В нынешнем сезоне он отыграл за команду 22 встречи, отметившись 18 голами.
- Его признали лучшим футболистом по итогам 2016 года.
- Примечательно, что в начале неделе СМИ признали самым дорогим футболистом мира бразильца Неймара – его оценили в 246 миллионов евро, тогда как Роналду занял только седьмую строку (126 млн евро).

}

В результате кластеризации коллекции из 2000 документов на пять кластеров было получено следующее условное распределение

документа x_1 по кластерам

$$p(t|x) = (\ 0.04;\ 0.26;\ 0.61;\ 0.09;\ 0.00\ )$$

Реферат №1.

1. $Q = 8$
2. Реферат строился относительно одного кластера, к которому документ относится с большей вероятностью, кластер №3.
3. Документ не разбивался на части.
4. Объем реферата V составлял 40% от оригинала, $V = 4$.

   При таких условиях реферат состоял из предложений №1, 3, 4, 6.

   ⟹ $R =$ (1,3,4,6)

- Роналду решили продать в китайский клуб за 300 млн евро.
- Руководство «Реала» перед таким предложением не устояло.
- Примечательно, что испанский клуб продлил контракт с португальцем только в ноябре прошлого года.
- Если сделка по продаже все же состоится, Роналду перейдет в китайский клуб в сезоне 2018/19.

Реферат №2.

1. $Q = 8$.
2. Кластер №3.
3. Документ не разбивался на части.
4. Объем реферата V зависел от разницы весов предложений (см. 3.6.4 пункт с), $W = (5.8; 6.6; 4.7; 8.2; 2.4; 7.2; 3.9; 6.7; 2.0; 5.1)$,

   $\varepsilon = 1, \delta = 4.1$

   ⟹ $R = (1, 2, 3, 4, 6, 8, 10\ )$

- Роналду решили продать в китайский клуб за 300 млн евро.
- Как сообщает ТАСС со ссылкой на испанскую прессу, китайцы готовы выложить за форварда 300 миллионов евро.
- Руководство «Реала» перед таким предложением не устояло.
- Примечательно, что испанский клуб продлил контракт с португальцем только в ноябре прошлого года.
- Если сделка по продаже все же состоится, Роналду перейдет в китайский клуб в сезоне 2018/19.
- В нынешнем сезоне он отыграл за команду 22 встречи, отметившись 18 голами.
- Примечательно, что в начале неделе СМИ признали самым дорогим футболистом мира бразильца Неймара – его оценили в 246 миллионов евро, тогда как Роналду занял только седьмую строку (126 млн евро).

Реферат №3.

1. $Q = 8$.
2. В соответствии с распределением $p(t|x)$ для выбранного документа 60% реферата строилось относительно кластера №3, 30% – кластера №2, 10% – кластера №4.
3. Документ не разбивался на части.
4. $V = 5$.

Относительно кластера №3 были выбраны предложения №1, 4, 6, 8; относительно кластера №2 – предложения №6, 10; относительно кластера №1 – предложение №4.

$$\Rightarrow \quad R = (1, 4, 6, 8, 10)$$

- Роналду решили продать в китайский клуб за 300 млн евро.
- Примечательно, что испанский клуб продлил контракт с португальцем только в ноябре прошлого года.
- Если сделка по продаже все же состоится, Роналду перейдет в китайский клуб в сезоне 2018/19.
- В нынешнем сезоне он отыграл за команду 22 встречи, отметившись 18 голами.
- Примечательно, что в начале неделе СМИ признали самым дорогим футболистом мира бразильца Неймара – его оценили в 246 миллионов евро, тогда как Роналду занял только седьмую строку (126 млн евро).

Реферат №4.

1. Аннотация строилась с помощью онлайн системы Рефератор–Visual World (URL: https://visualworld.ru/referat.jsp).
2. V составлял 40% от оригинала, $V = 4$.

$$\Longrightarrow \qquad R = (1, 2, 6, 10)$$

- Роналду решили продать в китайский клуб за 300 млн евро.
- Как сообщает ТАСС со ссылкой на испанскую прессу, китайцы готовы выложить за форварда 300 миллионов евро.
- Если сделка по продаже все же состоится, Роналду перейдет в китайский клуб в сезоне 2018/19.
- Примечательно, что в начале неделе СМИ признали самым дорогим футболистом мира бразильца Неймара – его оценили в 246 миллионов евро, тогда как Роналду занял только седьмую строку (126 млн евро).

3.9. Сравнение результатов реферирования

Для выборки из десяти новостных статей строились три разных реферата.

Реферат №1: строился с помощью алгоритма, реализованного в

рамках данной работы.

Реферат №2: составлялся человеком.

Реферат №3: определялся в результате работы онлайн системы Рефератор–Visual World (URL: https://visualworld.ru/referat.jsp).

В виду того, что человеческое мнение субъективно, в формировании рефератов принимало участие три эксперта. Каждый из них выделял наиболее важные предложения в текстах, а в окончательный реферат входили только те предложения, которые были отмечены хотя бы двумя из трех экспертов. Объем реферата составлял ~40-50% от оригинала.

Таблица 6. Получение «идеального» реферата

	Эксперт №1	**Эксперт №2**	**Эксперт №3**	**Итог**
1	1,3,4,7,9,10, 12,16	3,5–11,15,16	1,3,4,8, 10-13,15	1,**3**,**4**,7,8,9, **10**,11,12,15,16
2	1,5,6,7,14,15	1,2,3,5,6,7,14	1,5,7,9,14,15	**1,5**,6,**7,14,**15
3	1,4,6,8,10	1,2,6,7,8	1,3,6,8,9	**1,6,8**
4	1,2,4,6,10	1,6,7,8,9	1,2,4,6	**1**,2,4,**6**
5	2,3,4,5,6,8,10, 13,14,17,20	2,6,7,10,11,13, 14,17,19,20	1-5,8,10,13, 14,17,20	**2**,3,4,5,6,8,**10, 13,14,17,20**
6	1,3,4	1,3,4	1,2,5	**1,**3,4
7	2,4,5,8,10, 11,17,18,20	2,5,9,12,13,17, 18,19,20	1,2,4,5,6,8,9, 11,18,19	**2**,4,**5**,8,9,11,17, **18**,19,20
8	1,3,6,9,11,14,15	1,3,6,7,8, 9,15-17	1,2,7,11,14,15	**1**,3,6,7,9,11, 14,**15**
9	1,4,5,9,10,13	1,4,5,9,10, 11,13	1,4,7,8,10, 11,13	**1,4**,5,9,**10**, 11,**13**
10	2,3,5,8	2,3,4,8	1,2,3,5	**2**,**3**,5,8

В Таблице 6 представлены номера предложений, выбранные для реферата каждым экспертом, и итоговый реферат, который мы будем считать «идеальным».

Жирным шрифтом выделены номера предложений, которые вошли в реферат по согласию всех трех экспертов.

В Таблице 7 представлена сравнительная характеристика полученных рефератов. В столбцах 1, 2, 3 приведено количество (указано в скобках) и доля (%) совпавших предложений в двух сравниваемых рефератах. Столбец №1 описывает разницу между Рефератом №1 и Рефератом №2, столбец №2 – между Рефератом №2 и Рефератом №3, столбец №3 – между Рефератом №1 и Рефератом №3.

Таблица 7. Анализ результатов реферирования

	АЛГОРИТМ	**№1**	«ИДЕАЛЬНЫЙ» РЕФЕРАТ	**№2**	ОНЛАЙН-СИСТЕМА	**№3**
1	3,4,7,10, 13,16,18	(5) 55%	1,**3**,**4**,7-9, **10**, 11,12,15,16	(4) 44%	2,3,8,9,13, 14,16	(3) 40%
2	2,3,5,7,9, 11,14	(3) 46%	**1,5**,6,**7,14,**15	(4) 61%	1,2,5,7,8, 10,14	(5) 71%
3	2,6,8,9,10	(2) 50%	**1,6,8**	(1) 33%	4,6,9	(2) 50%
4	1,4,6,8,10	(3) 67%	**1**,2,4,**6**	(2) 57%	1,2,10	(2) 50%
5	1–4,6,8,9, 11,13,17	(7) 67%	**2**,3-6,8,**10, 13,14,17,20**	(5) 45%	1,2,6-9,13, 14,16,18,19	(6) 57%
6	1,2,5	(1) 33%	**1,**3,4	(2) 57%	1,2,4,5	(3) 86%

7	2,4-7,11-13,16,18,19	(6) 57%	**2**,4,**5**,8,9,11,17,**18**,19,20	(4) 44%	2,3,5,6,7,11,14,19	(7) 74%
8	1-3,5,6,9,12,13,15	(4) 47%	**1**,3,6,7,9,11,14,**15**	(3) 42%	1,6,11,12,13,16	(4) 53%
9	2,4,7,8,9,10,13	(4) 57%	**1,4**,5,9,**10**,11,**13**	(2) 40%	1,2,4	(2) 40%
10	2,3,5,8	(4) 100%	**2**,**3**,5,8	(3) 75%	1,2,5,8	(3) 75%
		58%		**50%**		**60%**

Рефераты, построенные описанным в данной работе алгоритмом, в среднем на 58% рефераты совпадали с «идеальным» рефератом. Также следует отметить, что наибольшее согласие присутствует между рефератами, построенными с помощью алгоритма и онлайн-системой (60% совпадений).

Заключение

В рамках данной выпускной квалификационной работы был разработан алгоритм для построения рефератов к текстовым документам. Также для решения задачи реферирования был реализован метод кластеризации Information Bottleneck, который справлялся с задачей кластеризации лучше метода К-средних.

Рефераты, построенные на основе данного алгоритма, имели больший процент совпадений с «идеальными» рефератами, нежели рефераты, полученные в ходе работы системы Рефератор–Visual World (58% против 50%). Такие результаты открывают хорошую перспективу на будущее. Имеет место предположение, что дальнейшие модификации алгоритма могут улучшить качество реферирования и данный алгоритм найдет свое практическое применение в жизни и будет весьма востребован и полезен.

Список литературы

1. Браславский П., Густелев В. Система автоматического реферирования новостных сообщений на основе машинного обучения. URL:http://rcdl.ru/doc/2007/paper_54_v1.pdf (дата обращения: 24.03.2018).
2. Кудряшов Б. Д. Теория информации. СПб:Питер, 2009. 320 с.
3. Маннинг К. Д., Рагхван П., Шютце Х. Введение в информационный поиск. М.: Вильямс, 2011. 528 с.
4. Морфологический анализатор pymorphy2. URL: http://pymorphy2.readthedocs.io (дата обращения: 10.04.2018).
5. Allahyari M., Pouriyeh S., Assefi M., Safaei S., Trippe E. D., Gutierrez J.B., Kochut K. Text Summarization Techniques: A brief survey// International Journal of Advanced Computer Science and Applications, Vol. 8, No. 10. 2017. P. 397-405.
6. Edmundson H. New methods in automatic abstracting // Journal of the ACM. Vol. 16, No 2. 1969. P. 264–285.
7. Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Second edition. New York: Springer. 2017. 745 p.
8. Kumar Y. J., Goh O. S., Basiron H., Choon N. H., Suppiah P. C. A Review on Automatic Text Summarization Approaches // Journal of Computer Science. Vol. 12, No. 4. 2016. P. 178-190.
9. Luhn H. The automatic creation of literature abstracts // In IBM Journal of Research and Development. Vol. 2, No 2. 1958. P. 159–165.
10. MirShojaee S. H., Masoumi B., Zeinali E. Biogeography-Based Optimization Algorithm for Automatic Extractive Text Summarization. // International Journal of Industrial Engineering & Production Research.

Vol. 28, No. 1. 2017. P. 75-84.

11. Neto J. L., Freitas A. A., Kaestner C. A. A. Automatic Text Summarization using a Machine Learning Approach. 2002. URL: https://www.cs.kent.ac.uk/people/staff/aaf/pub_papers.dir/SBIA-2002-Joel.pdf (дата обращения: 24.02.2018).
12. NLTK. URL:https://www.nltk.org/ (дата обращения: 10.04.2018).
13. Sinha A., Yadav A., Gahlot A. Extractive Text Summarization using Neural Networks// arxiv.org. 2018. Дата обновления: 27.02.2018. URL: https://arxiv.org/ftp/arxiv/papers/1802/1802.10137.pdf (дата обращения: 20.03.2018).
14. Slonim N. The Information Bottleneck: theory and applications. Doctor of Philosophy thesis. The Hebrew University. 2002. 157 p.
15. scikit-learn. URL: http://scikit-learn.org (дата обращения: 10.04.2018).
16. 8. Tishby N., Pereira F., Bialek W. The information bottleneck method// Proc. 37th Allerton Conference on Communication and Computation. 1999. P. 368–377.
17. Torres-Moreno J.M. Automatic text summarization. Edition 1. Wiley-ISTE. 2014. 320 p.
18. re — Regular expression operations. URL:https://docs.python.org/3/library/re.html (дата обращения: 10.04.2018).
19. Zhao and G. Karypis. Criterion functions for document clustering - experiments and analysis. Technical report. University of Minnesota, Department of Computer Science. 2001.
20. Text-Summarization. URL: https://github.com/NutsLyam/Text-Summarization