Impact Factor:					
ISRA (India)	= 3.117	SIS (USA)	= 0.912	ICV (Poland)	= 6.630
ISI (Dubai, UAE)	= 0.829	РИНЦ (Russia)	= 0.156	PIF (India)	= 1.940
GIF (Australia)	= 0.564	ESJI (KZ)	= 8.716	IBI (India)	= 4.260
JIF	= 1.500	SJIF (Morocco)	= 5.667	OAJI (USA)	= 0.350

SOI: 1.1/TAS DOI: 10.15863/TAS

International Scientific Journal

Theoretical & Applied Science

p-ISSN: 2308-4944 (print) **e-ISSN:** 2409-0085 (online)

Year: 2019 **Issue:** 06 **Volume:** 74

Published: 14.06.2019 **http://T-Science.org**

SECTION 4. Computer science, computer engineering and automation

QR – Issue QR – Article

Marina Vladimirovna Shkurina
Peter the Great St. Petersburg Polytechnic University
Master's Student
Institute of Computer Science and Technology

Oleg Yurievich Sabinin
Peter the Great St. Petersburg Polytechnic University
Candidate of Engineering Sciences, Docent
Institute of Computer Science and Technology

COMPARATIVE ANALYSIS OF EXTRACTIVE TEXT SUMMARIZATION METHODS FOR TEXTS IN RUSSIAN LANGUAGE

Abstract: *Each day the amount of data online grows, which leads to needing to present information in a more condensed form, and automatic text summarization can help with that. In this article the applicability of some of the most popular methods for extractive summarization to texts in Russian language is explored.*

Key words: *Automatic text summarization, extractive text summarization, natural language processing, text analysis, data mining.*

Language: *Russian*

Citation: Shkurina, M. V., & Sabinin, O. Y. (2019). Comparative analysis of extractive text summarization methods for texts in Russian language. *ISJ Theoretical & Applied Science, 06 (74)*, 164-169.

Soi: http://s-o-i.org/1.1/TAS-06-74-17 ***Doi***: crossref https://dx.doi.org/10.15863/TAS.2019.06.74.17

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ИЗВЛЕКАЮЩИХ МЕТОДОВ АВТОМАТИЧЕСКОГО АННОТИРОВАНИЯ ДЛЯ ТЕКСТОВ НА РУССКОМ ЯЗЫКЕ

Аннотация: *Каждый день объем данных в сети Интернет и в хранилищах данных растет, из-за чего возникает необходимость представлять информацию в более сжатом формате, и автоматическое аннотирование текста может помочь решить эту задачу. В этой статье исследована применимость наиболее популярных методов извлекающего аннотирования для текстов на русском языке.*

Ключевые слова: *Автоматическое аннотирование текста, извлекающее аннотирование текста, обработка естественного языка, анализ текста, анализ данных.*

Введение

Сегодня человеку доступны огромные объемы информации, находящиеся в сети Интернет или в различных хранилищах данных, и объем этих данных увеличивается с невероятной скоростью. Согласно отчету аналитической фирмы IDC, проспонсированному Seagate, объем данных в 2025 году достигнет отметки в 175 зеттабайт (для сравнения, в 2018 году объем данных составил 33 зеттабайта). [1] При таких условиях появляется острая необходимость сокращения объемов этой информации до коротких, ёмких аннотаций. Составление аннотаций вручную – это очень трудоемкая задача, к тому же составление таких аннотаций для всех существующих текстов просто невозможно. В связи с этим все больший интерес вызывает задача автоматического аннотирования текста.

Исследования в этой области начались еще в 1950-ые годы [2], но все еще нельзя сказать, что задача полностью решена, в силу сложности и неоднозначности естественного языка. Несмотря на это существуют популярные методы автоматического аннотирования, которые позволяют в большинстве случаев получить довольно хорошие аннотации, содержащие выдержки из исходного текста.

Большая часть исследований в данной области проводится для наборов данных, где тексты и аннотации представлены на английском

языке. При этом исследования для текстов на русском языке встречаются довольно редко.

В данной статье будут кратко рассмотрены существующие подходы к автоматическому аннотированию, более подробно будут рассмотрены методы, наиболее часто использующиеся для аннотирования текстов на английском языке. Для рассмотренных методов будет проведен сравнительный анализ с использованием текстов на русском языке.

Подходы к автоматическому аннотированию

Существует несколько классификаций подходов к автоматическому аннотированию, и в каждом случае учитываются различные особенности задачи.

На основе количества входных документов выделяют аннотирование одного документа и аннотирование массива документов. [3] Во втором случае более критичной становится проблема избыточности – из исходных документов нужно выбрать предложения таким образом, чтобы информация, присутствующая в нескольких документах, не повторялась в аннотации.

Также методы автоматического аннотирования можно разделить на основе цели дальнейшего использования аннотации. Здесь выделяют общее аннотирование и аннотирование по запросу. Для второго типа результатом будет аннотация, содержащая только ту информацию, которая соответствует заранее заданному запросу.

По языку методы делятся на монолингвальные, мультилингвальные и кросслингвальные. Когда язык исходного документа и аннотации совпадает, то метод относится к монолингвальным. Если исходный документ написан на нескольких языках, так же, как и аннотация, то метод является мультилингвальным. Метод относится с кросслингвальным, когда язык исходного документа и язык аннотации различаются.

Наиболее принципиальная и важная классификация, которая является предметом большей части исследований, - это классификация на основе способа построения текста. В ней выделяются две группы методов: извлекающие и генерирующие. Извлекающие методы составляют аннотацию из предложений, присутствующих в тексте, а генерирующие методы способны создавать новый текст, которого нет в исходном документе.

Рассматриваемые методы

Для анализа были выбраны извлекающие методы аннотирования, которые снискали наибольшую популярность для аннотирования англоязычных текстов.

Метод с применением TF-IDF

TF-IDF (Term Frequency Inverse Document Frequency) – статистическая мера, которую можно использовать для оценивания важности конкретного слова в контексте всего документа, входящего в общую коллекцию. [4]

Эта мера состоит из двух частей. TF или частота слова - это отношение количества вхождения конкретного термина к суммарному набору слов в исследуемом тексте. IDF или инвертированная частота документа – это инверсия частотности, с которой определенное слово фигурирует в коллекции текстов. Это на самом деле показывает, насколько важно рассматриваемое слово в рамках текста. Учёт IDF уменьшает вес часто и широко употребляемых слов, что позволяет увеличить вес более редких слов, которые с большей вероятностью будут репрезентативными для данного текста.Частота слова рассчитывается по формуле (1), инвертированная частота документа по формуле (2), а сама метрика TF-IDF по формуле (3).

$$TF = \frac{n_t}{n}, \quad (1)$$

где n_t - число вхождений слова t в документ,

n – общее число слов в документе.

$$IDF = \log\frac{d}{d_t}, \quad (2)$$

где d – общее число документов,

d_t - число документов, в которых встречается t.

$$TF - IDF = TF \times IDF, \quad (3)$$

Алгоритм с использованием TF-IDF выглядит следующим образом:

1. На основе некоторого массива документов для всех встречающихся в них слов рассчитывается метрика IDF.

2. В документе, для которого необходимо сгенерировать аннотацию, метрика TF-IDF рассчитывается только для существительных в предложении.

3. На основе полученных значений предложения сортируются по убыванию. В итоговую аннотацию берутся первые n предложений, где n задается пользователем.

TextRank и LexRank

TextRank [5] и LexRank [6] – это методы на основе использования графов, которые были разработаны примерно в одно время двумя независимыми группами. В основе обоих методов лежит алгоритм PageRank [7], изначально созданный компанией Google для определения «важности» веб-страницы.

PageRank предполагает, что чем более важна веб-страница, тем больше на нее будут ссылаться другие страницы. При этом учитывается как число ссылок, так и их качество – то есть насколько важна страница, которая ссылается на рассматриваемую веб-страницу.

PageRank страницы i рассчитывается по формуле(4).

$$PR(i) = \frac{1-d}{N} + d\sum_{j=1}^{N}\frac{PR(j)}{C(j)}, \quad (4)$$

где N – число вершин графа,

$PR(j)$ – значение PageRank страницы j, которая ссылается на i,

$C(j)$ – общее количество страниц, на которые ссылается j,

d – коэффициент затухания, который находится в диапазоне $[0;1]$ (в классической формуле принимается равным 0,85).

Для методов TextRank и LexRank процесс создания аннотация можно разделить на два этапа:

1. Создание графа: предложения текста являются вершинами, а степень сходства предложений – весами ребер, соединяющих эти предложения.

2. Реализация алгоритма PageRank с учетом весов: это позволяет извлечь из текста n предложений с самым высоким рангом.

В методе TextRank оценка степени сходства двух предложений S_i и S_j, которые представляют собой набор слов, которые в нем появляются ($S_i = w_1^i, w_2^i, \dots, w_{N_i}^i$), осуществляется по формуле (5).

$$sim(S_i, S_j) = \frac{|\{w_k | w_k \in S_i \& w_k \in S_j\}|}{\log(|S_i|) + \log(|S_j|)} \quad (5)$$

Метод LexRank для оценки степени сходства использует модифицированную косинусную меру, которая рассчитывается по формуле (6).

Здесь для оценки сходства используются метрики TF и IDF, которые были описаны ранее.

Метод с применением латентно-семантического анализа

$$sim(S_i, S_j) = \frac{\sum_{w \in S_i, S_j} tf_{w,S_i} tf_{w,S_j} (idf_w)^2}{\sqrt{\sum_{x_i \in S_i} (tf_{x_i,S_i} idf_{x_i})^2} \times \sqrt{\sum_{y_i \in S_j} (tf_{y_i,S_j} idf_{y_i})^2}} \quad (6)$$

Латентно-семантический анализ (ЛСА) – это вычислительная модель, которая позволяет представить семантику на основе следующих идей: два слова близки семантически, если появляются в схожем контексте, и два контекста схожи, если содержат семантически близкие слова. [8] Первый шаг латентно-семантического анализа – это создание матрицы слово-на-предложение $S_{[P\times N]}$. Строки матрицы соответствуют словам, а колонки – предложениям. В ячейки записывается, сколько раз данное слово встретилось в данном предложении.

После создания матрицы применяется инструмент сингулярного разложения (Singular Value Decomposition, SVD). Согласно теореме о сингулярном разложении, любая вещественная прямоугольная матрица может быть разложена на произведение трех матриц:

$$S = U\Sigma W^T, \quad (7)$$

где $U_{[P\times P]}$ – унитарная матрица,

$\Sigma_{[P\times N]}$ – матрица, у которой все элементы, не лежащие на диагонали равны нулю,

W^T – это матрица, которая является результатом транспонирования унитарной матрицы $W_{[N\times N]}$.

Диагональные элементы Σ являются отсортированными сингулярными значениями: $\Sigma_{[i,i]} > \Sigma_{[i+1,i+1]}$. Можно получить приближение матрицы S другой матрицей меньшего ранга k:

$$S \approx U\Sigma_k W^t = U_k \Sigma_k W_k^T, \quad (8)$$

где U_k и W_k соответствуют первым k столбцам U и W.

Такое разложение позволяет уменьшить размерность исходной матрицы, подчеркнуть наиболее сильные связи и при этом избавиться от шума.

Обычно в качестве k выбирают число от 100 до 500, но в целом выбор зависит от размера документа.

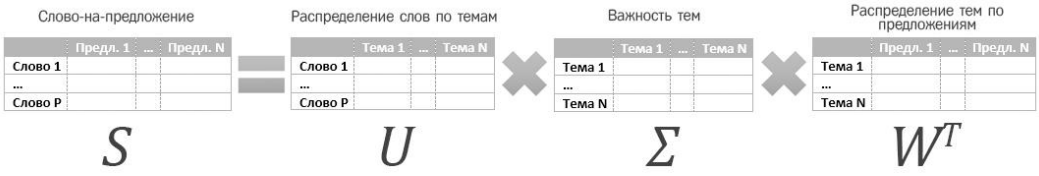

Рисунок 1 - Сингулярное разложение матрицы слово-на-предложение и его интерпретация

Последний шаг в методах ЛСА – это собственно выбор предложений для аннотации. Есть несколько вариаций методов ЛСА, которые отличаются способом отбора, но одни из лучших результатов показывает метод Cross [9].

Здесь также выполняется сингулярное разложение матрицы S, но после этого шага и перед выбором предложений проводится дополнительная обработка – для каждой темы, которая представлена строкой в векторе W^T, рассчитывается среднее значение. В ячейки этой строки, в которых значение меньше или равно среднему, записывается ноль. Этот шаг нужен, чтобы для каждой темы исключить предложения, которые не являются ключевыми.

Затем для каждого предложения рассчитывается так называемая длина: сначала каждая колонка, соответствующая предложению, умножается на сингулярные значения матрицы Σ. Это делается с целью придать больший вес более важным темам. После этого для каждой колонки суммируются все ее значения – эта сумма и называется длиной предложения. В аннотацию выбираются предложения с самым большим значением длины.

Оценка полученных результатов

Оценка аннотации – непростая задача, так как для конкретного документа или набора документов не существует идеальной аннотации.

На сегодняшний день для оценки систем автоматического аннотирования наиболее популярной и широко используемой является набор метрик ROUGE [10], предложенный в 2004 году и ставший стандартом автоматической оценки аннотаций. В этот набор входят метрики ROUGE-N, ROUGE-L, ROUGE-W и ROUGE-S.

Метрика ROUGE-N, которая чаще всего используется для оценки результатов, основывается на сравнении n-грамм (в качестве n чаще всего берется 1 или 2), полученных из набора эталонных аннотаций и n-грамм оцениваемой аннотации, и вычисляется по формуле (9):

$$ROUGE - N = \frac{\sum_{S \in R} \sum_{g_n \in S} C_m(g_n)}{\sum_{S \in R} \sum_{g_n \in S} C_m(g_n)}, \qquad (9)$$

где R – множество эталонных аннотаций;

g_n – n-грамм длины n;

$C_m(g_n)$ – количество n-грамм g_n, совпавших для эталонной и оцениваемой аннотации;

$C(g_n)$ – количество n-грамм g_n в эталонной аннотации.

По сути, ROUGE-N является метрикой, оценивающей полноту. То есть ее значение будет тем выше, чем больше информации, присутствующей в эталонной аннотации, вошло в оцениваемую аннотацию, при этом наличие в оцениваемой аннотации лишней информации не учитывается.

Таблица 1. Результаты оценки сравниваемых методов

Метод	ROUGE-1 Степень сжатия					ROUGE-2 Степень сжатия				
	0.9	**0.8**	**0.7**	**0.6**	**0.5**	**0.9**	**0.8**	**0.7**	**0.6**	**0.5**
Baseline	0.34	0.43	0.48	0.54	0.57	0.03	0.05	0.06	0.08	0.09
TF-IDF	0.44	0.52	0.55	0.59	0.62	0.06	0.08	0.09	0.11	0.12
TextRank	0.42	0.50	0.56	0.59	0.61	0.05	0.08	0.10	0.11	0.12
LexRank	0.36	0.45	0.51	0.57	0.60	0.04	0.06	0.08	0.10	0.12
LSA	0.47	0.54	0.58	0.62	0.64	0.07	0.09	0.11	0.13	0.13

Для русского языка крупной проблемой в области автоматического аннотирования является отсутствие размеченных наборов данных, на основе которых можно было бы обучать системы автоматического аннотирования или проверять их результаты. В связи с этим в качестве наборов данных использовались главы из произведений русской литературы, а в качестве эталонных аннотаций – краткие содержания этих глав, найденные в сети Интернет. Всего было использовано 11 текстов и аннотаций.

Для оценки результатов каждый алгоритм запускался несколько раз с разными параметрами степени сжатия (на основе степени сжатия

рассчитывается, сколько предложений должно быть в итоговой аннотации): от 50% сжатия до 90%. Для каждой полученной аннотации проводилось сравнение с эталонной аннотацией и вычислялись значения метрик ROUGE-1 и ROUGE-2.

Чтобы оценить, насколько хорошие результаты показывают рассматриваемые методы, был также введен baseline – простейший алгоритм, который формирует аннотацию выбором случайных предложений из текста. Введение такого baseline помогает понять, улучшаются ли генерируемые аннотации от внедрения более сложных вычислительных моделей, и если улучшаются, то насколько.

В таблице 1 приведены результаты для сравниваемых методов, полученные вычислением метрик ROUGE-1 и ROUGE-2 и усреднением результатов для разных текстов.

На основе полученных данных можно увидеть, что все сравниваемые методы показывают лучшие результаты, чем случайный выбор предложений. При этом графовые алгоритмы TextRank и LexRank показывают худшие результаты по сравнению с другими методами. Лучший результат показывает метод с использование латентно-семантического анализа.

Для всех методов наблюдается ухудшение результата при увеличении степени сжатия, что обоснованно, так как чем больше предложений в генерируемой аннотации, тем больше возможностей у метода выбрать информативное предложение.

Проблемы извлекающих методов аннотирования

С помощью извлекающих методов аннотирования можно добиться неплохих результатов, но такие методы сильны ограничены и не всегда сгенерированные аннотации корректны:

– Извлекающие методы могут только либо включить, либо исключить предложение из аннотации.

Если предложение очень длинное, и при этом содержит очень важную информацию, система аннотирования должна будет включить предложение целиком, даже если оно содержит много лишней информации. Это может сильно снизить эффективность автоматического аннотирования.

– Текст далеко не всегда получается связным.

Так как предложения могут быть взяты из разных частей текста, которые могут находиться далеко друг от друга, то читабельность полученного текста не всегда оказывается на высоком уровне. Если в аннотацию выбирается предложение, которое ссылается на информацию, представленную в одном из предыдущих, и это предыдущее предложение не включается в аннотацию, то читатель может не понять, о чем речь.;

– Текст аннотации может искажать факты.

Помимо того, что в аннотацию может войти предложение, для понимания которого будет не хватать другого предложения из исходного текста, включение таких предложений может исказить исходный смысл. Так, например, если есть несколько предложений: “Иван купил себе новые тапки. Кот Ивана, Мурзик, не оценил их по достоинству. На следующий день он изжевал эти тапки в клочья.” Если в аннотацию войдут первое и третье предложение, то читатель подумает, что это Иван изжевал собственные тапки в клочья. Для того, чтобы избежать такой проблемы, в системах аннотирования могут использоваться методы разрешения анафоры и кореферентности.

Заключение

В данной работе были рассмотрены существующие подходы к решению задачи автоматического аннотирования, были более подробно изучены наиболее популярные методы извлекающего аннотирования, а также было проведено сравнение эффективности работы этих методов для текстов на русском языке.

На основе проведенной работы можно сделать следующие выводы:

– с помощью методов извлекающего аннотирования можно получать достаточно неплохие результаты;

– среди рассматриваемых методов для имеющихся данных лучшие результаты показал метод латенто-семантического анализа;

– любая система извлекающего аннотирования имеет ограничения, и в некоторых случаях информация в аннотации может быть искажена.

Проведенное исследование показывает, что для получения более качественных аннотаций на основе методов извлекающего аннотирования необходимо избавляться от лишней информации в предложениях, сохранив при этом важную информацию (использование методов для сжатия предложений), а также снизить вероятность получения аннотации с некорректной информацией (использование методов с разрешением анафоры и кореферентности).

References:

1. Reinsel, D., Gantz, J., & Rydning, J. (2018). *The Digitization of the World From Edge to Core*.
2. Luhn, H. P. (1958). The Automatic Creation of Literature Abstracts. *IBM Journal of Research and Development, Vol. 2, № 2*, pp. 159–165.
3. Gambhir, M., & Gupta, V. (2017). Recent automatic text summarization techniques: *a survey Artificial Intelligence Review. Springer Netherlands, Vol. 47, № 1*, pp. 1–66.
4. Thanaki, J. (2017). *Python Natural Language Processing: Advanced machine learning and deep learning techniques for natural language processing*. (p.486). Packt Publishing.
5. Mihalcea, R., & Tarau, P. (2004). *TextRank: Bringing Order into Texts Conference on Empirical Methods in Natural Language Processing* , EMNLP 2004, A Meeting of Sigdat, A Special Interest Group of the Acl, Held in Conjunction with ACL 2004, 25-26 July 2004, Barcelona, Spain.
6. Erkan, G., & Radev, D. R. (2004). LexRank: Graph-based Lexical Centrality as Salience in Text Summarization. *Journal of Artificial Intelligence Research, Vol. 22*, pp. 457–479.
7. Page, L., et al. (1999). The PageRank Citation Ranking: Bringing Order to the Web World Wide Web Internet And Web Information Systems.
8. Torres-Moreno, J.-M. (2014). *Automatic Text Summarization*. Hoboken, NJ, USA: John Wiley & Sons, Inc.
9. Ozsoy, M., Cicekli, I., & Alpaslan, F. (2010). *Text summarization of Turkish texts using Latent Semantic Analysis.* Proceedings of the 23rd International Conference on Computational Linguistics, № August, pp. 869–876.
10. Lin, C. Y. (2004). *ROUGE: A Package for Automatic Evaluation of summaries* Conference: In Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004).