Электронный научный журнал "Математическое моделирование, компьютерный и натурный эксперимент в естественных науках" http://mathmod.esrae.ru/
URL статьи: **mathmod.esrae.ru/41-162**
Ссылка для цитирования этой статьи:
Лезгян А.С. Автоматическое реферирование текстов: классификация, архитектуры, современные подходы и проблемы // Математическое моделирование, компьютерный и натурный эксперимент в естественных науках. 2023. № 1.

УДК 004.855.5:539.3 DOI: 10.24412/2541-9269-2023-1-19-27

АВТОМАТИЧЕСКОЕ РЕФЕРИРОВАНИЕ ТЕКСТОВ: КЛАССИФИКАЦИЯ, АРХИТЕКТУРЫ, СОВРЕМЕННЫЕ ПОДХОДЫ И ПРОБЛЕМЫ

Лезгян А.С.[1]
[1]Саратовский государственный технический университет имени Гагарина Ю.А., Россия, Саратов, lezgyan@yandex.ru

AUTOMATIC TEXT SUMMARIZATION: CLASSIFICATION, ARCHITECTURES, MODERN APPROACHES AND CHALLENGES

Lezgyan A.S.[1]
[1]Yuri Gagarin State Technical University of Saratov, Russia, Saratov, lezgyan@yandex.ru

Аннотация. В статье рассматривается задача автоматического реферирования текстов. Приводится классификация существующих методов, описываются их достоинства и недостатки. В статье приведена общая архитектура подобных систем и ее модификации в зависимости от метода реферирования. Рассматриваются современные подходы, основанные на использовании глубоких нейронных сетей, их особенности, варианты использования, существующие проблемы и возможные пути их решения.

Ключевые слова: машинное обучение; обработка текстов на естественном языке; автоматическое реферирование, Transformer.

Abstract. This article focuses on the task of automatic text summarization. It provides a classification of existing methods, describing their strengths and weaknesses. The article presents a general architecture of such systems and its modifications based on the summarization method. It explores modern approaches based on deep neural networks, discussing their characteristics, application scenarios, existing challenges, and potential solutions.

Keywords: machine learning, natural language processing, automatic summarization, Transformer.

В наши дни растущий в геометрической прогрессии объем текстовой информации делает все более трудоемкой задачу обработки и анализа массивов данных. В таких условиях особенно актуальна возможность эффективного

обобщения текстов для последующего использования, а соответственно и задача автоматического реферирования.

Автоматическое реферирование заключается в создании краткого и лаконичного резюме с сохранением ключевой информации и общего смысла исходного текста. Резюме можно определить как некоторый текст, составленный на основе одного или нескольких исходных текстов и передающий основную информацию, содержащуюся в исходном тексте (текстах). При этом по размеру резюме составляет не более половины исходного текста (текстов), а обычно сильно меньше его. Автоматическое реферирование является сложной задачей, поскольку при обработке фрагмента текста требуется не только анализ его смыслового содержания и выделение наиболее важной информации, но и последующая генерация связного, соответствующего всем правилам естественного языка, реферата.

Методы автоматического реферирования имеют несколько различных классификаций, разнящихся от источника к источнику. Чаще всего выделяют три больших класса: по типу исходных данных, по результату и по цели реферирования. Подробнее остановимся на классификации по результату реферирования, относительно которой существует два основных подхода к обобщению текста: экстрактивный и абстрактивный. Методы экстрактивного реферирования основываются на извлечении некоторого подмножества наиболее важных предложений исходного текста. В отличие от них, методы абстрактивного реферирования интерпретируют и анализируют исходный текст, используя передовые методы обработки естественного языка, после чего генерируют новый, более короткий текст, содержащий основную информацию из оригинального источника.

Вне зависимости от используемого подхода, общая архитектура системы автоматического реферирования состоит из следующих этапов:

1. **предобработка:** в зависимости от используемых методов на первом этапе может производиться сегментация, токенизация, удаление стоп-слов, стемминг, лемматизация и т.д.;
2. **основная обработка:** предварительно подготовленный и очищенный текст обрабатывается одним или несколькими методами реферирования, преобразующими исходный документ в реферат;
3. **постобработка:** после этапа основной обработки может потребоваться дополнительная перестановка предложений или замена некоторых слов, что также повышает качество реферата.

Выбор конкретного подхода в основном влияет на этап основной обработки, так как конкретизирует какие алгоритмы будут использованы.

Несмотря на то, что рефераты, создаваемые людьми, обычно являются абстрактивными, существенная доля исследований посвящена экстрактивному реферированию. Это объясняется относительной простотой и производительностью подобных решений, а также довольно высокой точностью, поскольку грамотность и корректность терминологии в данном

случае зависят только от исходного текста. Однако данные методы обладают и рядом существенных недостатков, а именно: избыточностью извлекаемых предложений, отсутствием временной и смысловой согласованности между извлеченными предложениями, в особенности при реферировании нескольких текстов, «размазанность» информации по нескольким предложениям и возможные противоречия, наблюдающиеся в текстах, которые содержат более одной основной темы.

Обработка текста при экстрактивном реферировании обычно включает в себя:

1. Создание внутреннего представления текста, удобного для дальнейшего анализа, такие как мешок слов, векторные представления, скрытые состояния кодировщиков нейросетевых моделей, графы и т.д.;
2. Взвешивание предложений на основе внутреннего представления;
3. Извлечение предложений с наиболее высокой оценкой с последующим их объединением в реферат.

Абстрактивный подход позволяет реферировать тексты аналогично тому, как это делает человек, и, соответственно, является более перспективным. Благодаря тому, что перефразирование, гибкая генерация и слияние информации решают проблемы несогласованности и избыточности, присущие экстрактивному реферированию, в результате получаются более качественные и краткие рефераты. Однако абстрактивный подход сложнее, медленнее и требует больших вычислительных ресурсов. Кроме того, требуется не только глубокое понимание смысла исходного текста, но и механизм генерации, позволяющий грамотно и в сжатом виде пересказывать основную информацию. Это осложняется тем, что задача **генерации естественного языка** сама по себе является одной из открытых и до конца не решенных проблем. Методы генерации текста обладают такими недостатками как повторение одинаковых слов, «галлюцинации», проблемы с обработкой слов, отсутствующих в словаре модели (OOV) и т.д.

Процесс абстрактивного реферирования обычно состоит из создания внутреннего семантического представления исходного текста и последующей генерации реферата с использованием языковой модели.

Развитие глубоких нейронных сетей показало их эффективность в задачах автоматической обработки текстов. Ключевыми этапами развития нейронных сетей в качестве основного инструмента автоматического реферирования можно считать: появление рекуррентных нейронных сетей LSTM [2] и GRU [3], изобретение механизма внимания [4], и, наконец, разработка основанной на внимании архитектуры Transformer [5].

Чаще всего задача автоматического реферирования решается при помощи архитектуры seq2seq [6]. В общем виде она состоит из кодировщика (Encoder), который позволяет анализировать исходный текст и сохранять информацию о нем во внутренних скрытых состояниях, и декодировщика (Decoder), который служит для генерации результирующей последовательности, основываясь на

информации из скрытых состояний кодировщика. В качестве кодировщика и декодировщика могут выступать различные модели, пригодные для обработки последовательностей данных.

Долгое время в качестве кодировщиков и декодировщиков использовались рекуррентные нейронные сети (RNN). Главной особенностью RNN было наличие основывающейся на внутренних скрытых состояниях памяти, которая позволяла сохранять некоторый контекст во время обработки последовательностей данных переменной длины. Однако обычные RNN оставались подвержены проблеме анализа долгосрочных зависимостей, которая заключалась в том, что нейронная сеть начинала постепенно забывать контекст из начала предложения, анализируя каждое следующее слово. В качестве решения этой проблемы была разработана долгая краткосрочная память (LSTM) [2], а позднее и более производительные управляемые рекуррентные блоки (GRU) [3]. Рекуррентный блок был доработан путем добавления к нему памяти, реализованной как состояние ячейки, и набора фильтров, позволяющих определять, какую информацию следует сохранить как состояние ячейки, а какую можно «забыть».

Дальнейшее развитие seq2seq архитектуры связано с разработкой механизма внимания [4], позволяющего декодировщику использовать информацию не только из последнего слоя кодировщика, но и со всех предыдущих. Распределение внимания можно рассматривать как распределение вероятностей по словам исходной последовательности, которое показывает декодировщику, какую часть исходной последовательности нужно проанализировать, чтобы сгенерировать следующее слово. Модели seq2seq с механизмом внимания долгое время применялись как для абстрактивного так и для экстрактивного автоматического реферирования. Особое внимание следует уделить работе [7], в которой описывается гибридная архитектура указатель-генератор (Pointer-Generator). Реферат, как и в стандартной seq2seq модели, генерируется при помощи декодировщика, однако редкие или отсутствующие в словаре модели слова (OOV) копируются из исходного текста при помощи механизма указывания. Для этого на каждом шаге декодировщика вычисляется вероятность генерации, которая используется для выбора между генерацией слова из словаря модели на основе его вероятности или копирования из исходного текста на основе распределения внимания.

Как и рекуррентные нейронные сети, модели на основе архитектуры Transformer предназначены для обработки последовательностей данных, таких как текст на естественном языке, а соответственно могут использоваться для решения таких задач, как машинный перевод и автоматическое реферирование [5]. В отличие от RNN, архитектура Transformer не требует последовательной обработки данных, что позволяет распараллеливать обучение модели и обрабатывать большие объемы данных. В основе архитектуры Transformer лежит механизм self-attention, иногда называемый внутренним вниманием, который позволяет модели во время генерации обращаться к различным частям

исходного текста. Кроме этого, для лучшего понимания контекста каждого слова для вычисления внимания используется не одна общая матрица весов, а 8 независимо проинициализированных «голов» внимания, которые параллельно применяются к исходному тексту, после чего результат их работы объединяется. На основе архитектуры Transformer были созданы практически все современные модели, применяемые в задаче автоматического реферирования. Это могут быть как отдельные элементы исходной архитектуры, такие как генеративная языковая модель GPT [8] или маскированная языковая модель BERT [9], так и модификации всей исходной архитектуры, например, BART [10].

Главным недостатком архитектуры Transformer является ее вычислительная сложность. Механизм внутреннего внимания вычисляет зависимости между всеми словами исходного текста, а соответственно имеет квадратичную $O(n^2)$ сложность. Кроме того, подобные модели требуют большого количества размеченных обучающих данных, что так же является серьезной проблемой. В качестве решения был разработан подход переноса знаний (transfer learning), позволяющий обучать модель для решения одной задачи, а после дообучать ее для решения другой, не обязательно схожей, проблемы. В настоящее время наилучших результатов в автоматическом реферировании удалось достичь благодаря использованию предварительно обученных моделей BERT (модифицированный кодировщик Transformer) и GPT (модифицированный декодировщик Transformer). Данные модели обучаются языковому моделированию, что позволяет использовать в качестве обучающих данных большие объемы неразмеченных текстов.

Как уже было сказано, модель BERT является усовершенствованным многослойным двунаправленный кодировщиком из оригинальной архитектуры Transformer. В качестве задачи для предварительного обучения BERT используется маскированное языковое моделирование (MLM). Задача маскированного языкового моделирования состоит в предсказании распределения i-того токена по его левому и правому контекстам $P(w_i \mid w_1, \ldots, w_{i-1}, w_{i+1}, \ldots, w_n)$. Для этого часть токенов исходного текста заменяется на специальный токен [MASK], после чего модель обучается восстановлению скрытых токенов. Подобных подход не только улучшает способность модели к пониманию контекста каждого слова, но и позволяет использовать механизм внимания для всей обрабатываемой текстовой последовательности, а не только для левостороннего контекста, как в классических языковых моделях. В задаче экстрактивного реферирования модель BERT используется в качестве основы для построения классификатора, позволяющего определять наиболее важные предложения исходного текста [11]. При абстрактивном реферировании обычно применяется архитектура seq2seq в которой BERT используется в качестве кодировщика [11-12].

Модели GPT, в свою очередь, основываются на стандартном

декодировщике оригинального Transformer. Модели этого семейства обучаются стандартному или авторегрессионному языковому моделированию, то есть предсказанию следующего токена в последовательности с учетом предшествующих ему токенов $P(w_n \mid w_1, \ldots, w_{n-1})$. Хотя архитектура GPT не содержит кодировщика и предполагает только генерацию текста, существуют подходы, позволяющие использовать ее для широкого круга задач [8], в том числе и для абстрактивного автоматического реферирования. Для этого исходный текст $x = (x_1, \ldots, x_n)$ и целевой реферат $y = (y_1, \ldots, y_m)$ из обучающих данных конкатенируются через некоторый символ-разделитель δ в единый текст $w = (x_1, \ldots, x_n, \delta, y_1, \ldots, y_m)$, после чего модель дообучается все той же задаче языкового моделирования, при этом токены $(x_1, \ldots, x_n)$ и δ не учитываются во время вычисления функции потерь [13].

Дальнейшим развитием моделей на основе архитектуры Transformer стали гибридные подходы такие BART [10], сочетающий преимущества BERT и GPT.

Однако описанные выше архитектуры и подходы к обучению не решают одну из главных проблем автоматического реферирования, заключающуюся в обработке длинных текстов. Контекстным окном называют количество токенов, единовременно обрабатываемых моделью. Токен представляет собой языковую единицу, которая в зависимости от задачи может быть буквой, словом или n-граммой. BERT использует контекстное окно равное 512 токенам, максимальный размер окна у GPT достигает 2048 токенов. Это обусловлено тем, что, как было сказано ранее, механизм внимания, использующийся в этих моделях, обладает квадратичной вычислительной сложностью. Более длинные текстовые последовательности существенно замедлят вывод нейронной сети. Возможные способы решения данной проблемы можно разделить на две большие группы: методы, предполагающие обработку текста по частям с использованием уже существующих моделей, и методы, предполагающие возможность снижения вычислительной сложности механизма внимания.

Возможность обработки текста в несколько этапов, с применением как одной, так и нескольких моделей, рассмотрены в работах [14-15]. В работе [14] предлагается использование подхода, схожего с тем, как люди реферируют длинные тексты. Человек должен прочитать исходный текст, понять его содержание, выделить наиболее важные части, и пересказать их в сжатом виде. Для имитации подобного поведения предполагается использовать бинарный классификатор на основе модели BERT, предварительно обученный выделять наиболее важные предложения исходного текста. Такой подход позволяет сжимать исходный текст в среднем на 61%. После этого сжатый текст подается на вход абстрактивной модели, в работе использовалась предварительно обученная модель BART, которая и генерирует итоговое резюме. В работе [15] разработчики архитектуры GPT предлагают альтернативный подход, позволяющий с определенной точностью реферировать книги. Авторы используют алгоритм, основанный на рекурсивной декомпозиции исходного

текста и последующем обобщении его небольших частей. При решении сложной задачи ее часто разбивают на более простые подзадачи, решение которых позволит впоследствии разобраться с исходной проблемой. Для декомпозиции текста авторы используют достаточно простую процедуру: короткие части текста реферируются целиком, более длинные разбиваются на несколько мелких фрагментов и рекурсивно обрабатываются. Для того, чтобы сохранить связь между отдельными частями исходного текста и не потерять контекстную информацию, авторы дополнительно помещают предыдущие отреферированные блоки текста в контекст к следующему обрабатываемому блоку.

Главным недостатком упомянутых подходов является частичная потеря контекстной информации. Предварительное сжатие текста бинарным классификатором может привести к потере важной информации при ложноположительном (FP) срабатывании. Рекурсивное реферирование не гарантирует сохранения связи между разными блоками текста, в особенности находящимися далеко друг от друга.

Альтернативный подход заключается в снижении вычислительной сложности механизма внимания с квадратичной $O(n^2)$ до линейной $O(n)$. В работе [14] представлена архитектура Longformer, главная идея которой заключается в замене глобального механизма внутреннего внимания между всеми токенами на комбинацию из оконного внимания в локальном контексте и глобального внимания для определенных служебных токенов. Подобная архитектура применима как для классических, так и для маскированных языковых моделей. Для анализа локального контекста используется скользящее окно из w токенов. Использование многослойной архитектуры с подобным механизмом внимания позволяет охватить большую часть всей информации. Верхние слои имеют доступ ко всем входным данным по аналогии со сверточными нейронными сетями. Вычислительная сложность подобного подхода для последовательности длины n и контекстного окна размерности w составляет $O(n \times w)$ и линейно масштабируется в зависимости от длины входной последовательности. Однако для некоторых задач, к примеру классификации или ответов на вопросы (QA), этот механизм может оказаться недостаточно гибким. Поэтому, в дополнение к оконному вниманию, используется и стандартное глобальное внимание, примененное, однако, лишь к некоторым токенам. Для задачи классификации это токен конца последовательности [CLS], а для QA – все токены вопроса. Так как количество подобных токенов относительно длины текста невелико и не зависит от нее, сложность комбинированного локального и глобального внимания по-прежнему составляет $O(n)$. И, хотя выбор служебных токенов специфичен для каждой задачи, подобный подход избавляет от необходимости разработки специфичных архитектур. Позднее исследовательская группа Google представила архитектуру Big Bird [15], сочетающую в себе как наработки из

[14], так и собственные улучшения механизма внимания путем добавления в него компонента случайного внимания. Помимо этого, в работе приводится строгие теоретические обоснования работоспособности комбинированного механизма внимания, который до этого разрабатывался в основном эмпирически.

Таким образом можно заключить, что благодаря развитию глубокого обучения за последние годы в задаче автоматического реферирования наблюдается серьезный прогресс. При этом остаются открытыми задачи оптимизации производительности, особенно актуальные в связи с увеличением размерности языковых моделей, а также задачи анализа длинных текстовых последовательностей, в которых существующие методы все еще серьезно уступают по качеству человеческим результатам.

Литература

1. Radev, D.R., Hovy, E.H., McKeown, K. Introduction to the Special Issue on Summarization // Computational Linguistics. 2002. № 28 (4). С. 399-408.
2. Hochreiter, S., Schmidhuber, J. Long Short-Term Memory // Neural Computation. 1997. № 9. С. 1735-1780.
3. Cho, K., Merrienboer, B.V., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation // Conference on Empirical Methods in Natural Language Processing. 2014. С. 1724–1734.
4. Brauwers G., Frasincar F. A General Survey on Attention Mechanisms in Deep Learning // IEEE Transactions on Knowledge and Data Engineering. 2023. № 35 (4). С. 3279-3298.
5. Vaswani, A., Shazeer, N.M., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I. Attention Is All You Need // 31st Conference on Neural Information Processing Systems (NIPS). 2017. С. 5998-6008.
6. Sutskever, I. Sequence to Sequence Learning with Neural Networks // Proceedings of the 27th International Conference on Neural Information Processing Systems. 2014. С. 3104–3112.
7. See A., Liu P., Manning C. Get To The Point: Summarization with Pointer-Generator Networks // Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. 2017. С. 1073–1083.
8. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T.J., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D. Language Models are Few-Shot Learners [Электронный ресурс] // arXiv.org. 2020. URL: https://arxiv.org/abs/2005.14165 (дата обращения: 25.04.2023).

9. Devlin, J., Chang, M., Lee, K., Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding // arXiv.org. 2019. URL: https://arxiv.org/abs/1810.04805 (дата обращения: 02.05.2023).
10. Lewis, M. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension // Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020. С. 7871–7880.
11. Liu, Y., Lapata, M. Text Summarization with Pretrained Encoders // arXiv.org. 2019. URL: https://arxiv.org/abs/1908.08345 (дата обращения: 05.05.2023).
12. Zhang, H., Gong, Y., Yan, Y., Duan, N., Xu, J., Wang, J., Gong, M., Zhou, M. Pretraining-Based Natural Language Generation for Text Summarization // Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL). 2019. С. 789-797.
13. Liu, P.J., Saleh, M., Pot, E., Goodrich, B., Sepassi, R., Kaiser, L., Shazeer, N.M. Generating Wikipedia by Summarizing Long Sequences // International Conference on Learning Representations (ICLR). 2018.
14. Beltagy, I., Peters, M.E., Cohan, A. Longformer: The Long-Document Transformer // arXiv.org. 2020. URL: https://arxiv.org/abs/2004.05150 (дата обращения: 05.05.2023).
15. Zaheer, M., Guruganesh, G., Dubey, K.A., Ainslie, J., Alberti, C., Ontañón, S., Pham, P., Ravula, A., Wang, Q., Yang, L., Ahmed, A. Big Bird: Transformers for Longer Sequences // Advances in Neural Information Processing Systems. 2020. № 33. С. 17283-17297.