КОМПЬЮТЕРНОЕ ОБЕСПЕЧЕНИЕ И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА

COMPUTER ENGINEERING AND SOFTWARE

Научная статья
УДК 004.912
https://doi.org/10.24143/2072-9502-2023-1-25-35
EDN EGRKGN

Анализ алгоритмов и решений для автоматической генерации подводок новостных статей в соцсетях с использованием искусственного интеллекта

А. И. Егунова, Р. С. Комаров, Ю. С. Вечканова, О. И. Егунова, Д. П. Сидоров, С. Д. Шибайкин, В. В. Никулин ✉

Национальный исследовательский Мордовский государственный университет им Н. П. Огарева, Саранск, Россия, nikulinvv@mail.ru ✉

Аннотация. При публикации статей в социальных сетях редакциям новостных порталов необходимо сформировать краткий реферат каждой статьи, затратив на это минимум времени. Оперативному и одновременному размещению публикации на всех зарегистрированных ресурсах способствует автоматическая генерация подводок. Предлагается использование алгоритмов искусственного интеллекта, обученных на корпусах русских текстов. Известны три подхода к реферированию текста для автоматизированного формирования подводок статей: экстрактивный, абстрактивный и комбинированный. Проводится сравнительный анализ методов экстрактивного и абстрактивного подходов в рамках решения задачи автоматической генерации подводок с помощью применения нейросетевых моделей машинного обучения. Проанализированы различные этапы экстрактивного реферирования с помощью как простых, так и более сложных методов: LexRank, TextRank и на основе Deep Learning. Путем сравнения выбраны абстрактивные модели как наиболее подходящие для выполнения суммаризации новостных статей, на основе модификации модели BERT. Более сложные генерирующие тексты обрабатывают тексты параллельно, что ускоряет обработку, но требует предобучения на больших корпусах новостных документов. При использовании абстрактивных моделей Pointer General Network и MBART сокращается время обработки информации, повышается эффективность работы.

Ключевые слова: суммаризация, реферирование, вектор, токен, кодирование, декодирование, генерация

Для цитирования: *Егунова А. И., Комаров Р. С., Вечканова Ю. С., Егунова О. И., Сидоров Д. П., Шибайкин С. Д., Никулин В. В.* Анализ алгоритмов и решений для автоматической генерации подводок новостных статей в соцсетях с использованием искусственного интеллекта // Вестник Астраханского государственного технического университета. Серия: Управление, вычислительная техника и информатика. 2023. № 1. С. 25–35. https://doi.org/10.24143/2072-9502-2023-1-25-35. EDN EGRKGN.

Original article

Analyzing algorithms and solutions for automatic generation of news article leads in social networks by using artificial intelligence

A. I. Egunova, R. S. Komarov, Yu. S. Vechkanova, O. I. Egunova, D. P. Sidorov, S. D. Shibaikin, V. V. Nikulin ✉

National Research Ogarev Mordovia State University, Saransk, Russia, nikulinvv@mail.ru ✉

Abstract. The article highlights the approaches to automatic abstracting the articles. When publishing articles on social networks, the editors of news portals need to create a short abstract of each article spending a minimum of time. Prompt and simultaneous placement of the publications on all registered resources is facilitated by automatic generation of leads. There is proposed to apply the artificial intelligence algorithms trained on corpora of the Russian texts. There are three approaches to text abstracting for the automated formation of article leads: extractive, abstract, and combined. There is carried out comparative analysis of the methods of extractive and abstract approaches in the frames of solving the problem by using neural network models of machine learning. Different stages of extractive abstracting are analyzed using both simple and more complex methods of LexRank, TextRank and on top of Deep Learning. The compared abstract models were selected as the most suitable ones for abstracting the news articles on top of the BERT model. More complex generating texts process the data in parallel, which speeds up processing, but requires training on large corpora of news documents. When using the abstract models Pointer General Network and MBART the information processing time is reduced and work efficiency increases.

Keywords: summarization, abstracting, vector, token, encoding, decoding, generating

For citation: Egunova A. I., Komarov R. S., Vechkanova Yu. S., Egunova O. I., Sidorov D. P., Shibaikin S. D., Nikulin V. V. Analyzing algorithms and solutions for automatic generation of news article leads in social networks by using artificial intelligence. *Vestnik of Astrakhan State Technical University. Series: Management, Computer Science and Informatics*. 2023;1:25-35. (In Russ.). https://doi.org/10.24143/2073-5529-2023-1-25-35. EDN EGRKGN.

Введение

Ежедневно новостные порталы с целью повышения трафика сайта распространяют статьи в социальных сетях. Перед публикацией главному редактору газеты необходимо изучить каждую статью, на что уходит много времени, т. к. газета формирует публикации за неделю с включением как тематических, так и оперативных новостей. Размер статьи варьирует от 300 до 1 500 слов, статьи размещаются не только на сайте газеты, но и в социальных сетях и мессенджерах. Разные сервисы под анонсы статей выделяют разный объем, учитывая который нужно сформировать краткий реферат каждой статьи. Решение задачи автоматической генерации подводок существенно ускорит подготовку издания и позволит оперативно размещать публикации на всех зарегистрированных ресурсах одновременно. Для выполнения данной операции, учитывая небольшой штат сотрудников районной газеты, планируется использовать алгоритмы искусственного интеллекта, обученные на корпусах русских текстов.

Основные причины, по которым автоматическое выделение релевантной информации текста может быть полезно:

– сокращается время обработки информации;

– повышается эффективность работы;

– алгоритмы автоматической суммаризации менее предвзяты, чем люди.

Существует 3 основных подхода к решению данной задачи: экстрактивный, абстрактивный и комбинированный.

Экстрактивный подход заключается в выделении из статьи наиболее значимых блоков информации. Блок может представлять собой набор предложений, абзацев или ключевых слов. Задачей экстрактивного подхода можно считать бинарную классификацию данных. Схема данного типа суммаризации представлена на рис. 1.

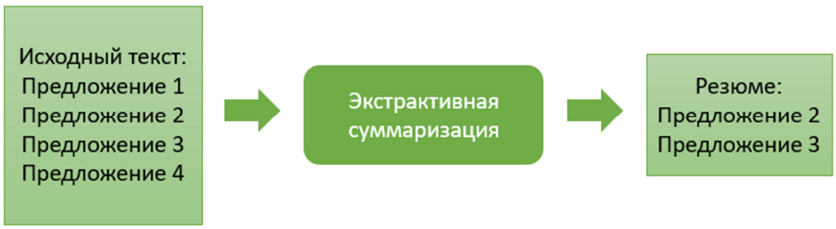

Рис. 1. Схема экстрактивного подхода суммаризации

Fig. 1. Graph of the extractive summarization approach

Методы данного подхода основаны на сортировке блоков информации по степени важности с помощью оценочной функции.

Абстрактивный подход значительно отличается от экстрактивного и заключается в генерации краткого содержания с порождением нового текста, содержательно обобщающего первичный документ. Модели данного подхода могут генерировать уникальное резюме, которое может содержать слова, отсутствующие в исходном тексте.

Под комбинированным подходом понимается совместное использование экстрактивного и абстрактивного подходов для решения одной задачи.

Экстрактивные подходы

Экстрактивное реферирование на основе вхождения общих слов. Этот метод считается наиболее простым, потому что происходит обработка только исходного текста. На первом шаге исходный текст раскладывается на предложения, а каждое предложение, в свою очередь, разбивается на токены (слова). Для каждого токена проводится лемматизация. В результате алгоритм сможет объединить одинаковые по смыслу слова в разных формах.

Следующим этапом, на основе функции схожести, которая вычисляется как отношение количества общих слов в двух предложениях к суммарной длине этих предложений, определяются коэффициенты схожести для каждой пары предложений [1].

Предложения, которые имеют общие слова, оставляют, и строится граф, в котором вершины представляют собой предложения статьи, а ребра показывают наличие одинаковых слов с соответствующими коэффициентами (весами) из предыдущего пункта алгоритма. В примере, представленном на рис. 2, отсутствует предложение 1, т. к. оно не имеет общих значимых слов с другими предложениями.

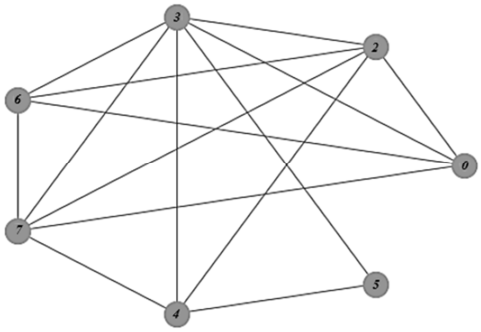

Рис. 2. Пример графа предложений

Fig. 2. Exemplary sentence graph

После сортировки (ранжирования) предложений по их значимости необходимо выбрать несколько самых значимых предложений и расположить их в соответствии с порядком их появления в исходном тексте.

Экстрактивное реферирование на основе обученных векторных представлений. Методы LexRank и TextRank строят из предложений граф, основываясь на их сходстве (например, косинусное сходство) [2]. Пример такого графа представлен на рис. 3.

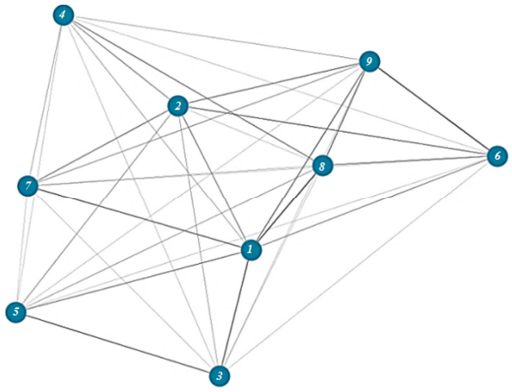

Рис. 3. Пример графа предложений, основанного на их сходстве

Fig. 3. Exemplary sentence graph based on the sentence similarity

Берется определенное количество блоков текста (например, предложений) и вычисляется матрица сходства. Сходства можно определить как веса в некотором графе. Ранжирование основывается на центральности вершин в графе. Текст новости токенизируется и лемматизируется и представляется в *tf-idf* форме [3]. Вычисляется матрица косинусов, которая определяет вероятность перехода из одной вершины в другую.

Центральность во взвешенном графе можно вычислить:

– через степень вершины (сумма весов всех ребер, которые соединены с вершиной):

$$D(i) = \sum_{j \in \text{neighbors}(i)} w_{ij},$$

где $D(i)$ – степень i-й вершины; w_{ij} – вес между i-й и j-й вершинами графа в матрице весов, если вершины i и j не смежные, то $w_{ij} = 0$;

– через PageRank – рекурсивную метрику, определяющую центральность вершины через центральность соседних вершин, где «важность» i-го предложения определяется по формуле

$$PR(p_i) = \frac{1-d}{N} + d \sum_{p_j \in M(p_i)} \frac{PR(p_j)}{L(p_j)},$$

где $PR(p_i)$ – PageRank p_i-го предложения; $PR(p_j)$ – PageRank j-го предложения; p_i – i-е предложение; p_j – j-е предложение; d – коэффициент демпфирования (по умолчанию 0,85); N – количество предложений; $M_{(pi)}$ – набор всех входящих ссылок на предложение p_i; $L_{(pj)}$ – количество исходящих ссылок на предложение p_j.

Данный метод в качестве вершин графа берет предложения из текста, а веса ребер вычисляются по формуле «схожести» p_i предложений по набору слов. PageRank можно представить в виде модели случайного серфера, который изначально находится в произвольной вершине a. Далее с некоторой вероятностью $P_{\text{initial}}(b|a)$ он может переместиться в другую вершину. Чтобы вероятности не были нулевыми, используется дополнительный коэффициент d демпфирования (обычно 0,85). С вероятностью $(1 - d) / N$ серфер перемещается в случайную вершину. Заранее выделенная вероятность перехода из одной вершины в соседнюю вычисляется нормализацией матрицы по строкам:

$$P(j \mid i) = \frac{1-d}{N} + dP_{\text{initial}}(j \mid i).$$

Вершины графа обозначают возможные состояния системы. Не зная начального состояния системы, можно вычислить вероятность того, что серфер находится в вершине j в текущий момент, умножив сумму вероятностей его нахождения в каждой из возможных вершин в момент времени $t - 1$ на вероятность перехода в вершину j из вершины i:

$$P(x_t = j) = \sum_i P(x_{t-1} = i) P(j \mid i),$$

где x_t – положение серфера в момент времени t; x_{t-1} – положение серфера в момент времени $t - 1$; $P(x_{t-1} = i)$ – вероятность, что в момент времени $t - 1$ серфер находится в вершине i; $P(j|i)$ – вероятность перехода из вершины i в вершину j.

В матричной форме эту формулу можно представить в форме стационарного распределения цепи Маркова [4]. Значения стационарного распределения $\vec{q}$ и будут PageRank'ами вершин:

$$\vec{q} = P^T \vec{q}.$$

После вычисления выбирается несколько предложений с наибольшими значениями для включения в реферат.

Экстрактивное реферирование на основе Deep Learning. Перед подачей в нейронную сеть текст необходимо токенизировать. Токенизация разбивает текст на значимые элементы. Каждый токен необходимо закодировать числом. Для этого перед обучением строят словарь (*mapping*) уникальных токенов, идентифицируя каждый из них уникальными числами (индексы). Словарь ограничен по длине (несколько десятков тысяч токенов).

В настоящее время популярна токенизация, разделяющая текст не на слова, а на часто встречающиеся их части (*byte-pair encoding*, *WordPiece*, *SentencePiece*). Это позволяет сжать много различных слов в относительно малое количество токенов. В самой нейросети для каждого токена в словаре используется некоторый вектор, который называется плотным, потому что имеет небольшую размерность относительно размерности словаря. Этот вектор обычно обучается вместе с сетью под конкретную задачу.

Сети-кодировщики текста (рис. 4) принимают на каждом слое на вход последовательность векторов y_i токенов x_i и преобразуют ее в новую последовательность.

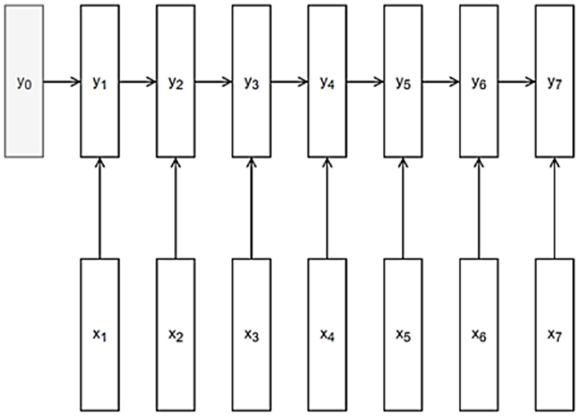

Рис. 4. Схема сети-кодировщика текста

Fig. 4. Graph of text encoder network

Все последовательности, кроме входной, считаются состоящими из «контекстуализированных» векторов. Способы сделать кодировщик: CNN, RNN, BiRNN, Transformer.

Сеть-трансформер (рис. 5) на каждом слое вычисляет представления токенов x_i, учитывая представления (вектора y_i) всех токенов на предыдущем слое.

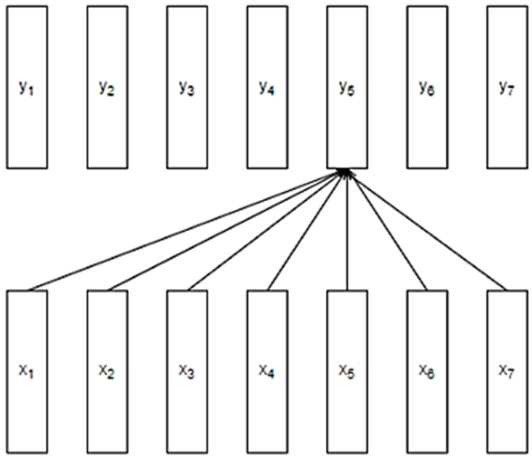

Рис. 5. Схема сети-трансформера текста

Fig. 5. Graph of network- transformer of the text

Сеть-декодировщик (*decoder*) – это сеть, используемая для генерации токенов. Авторегрессионные декодеры на шаге генерации (в инференсе) принимают на вход последовательность векторов уже сгенерированных токенов, кодируют ее (аналогично кодировщику) и выдают распределение вероятности над следующим словом: $P(w_t|w_{1:t-1})$.

Seq2Seq-архитектуры используются для генерации выходной последовательности с учетом входной, состоят из кодировщика исходного текста и декодера, который использует выход кодировщика как вспомогательный контекст. Такие архитектуры популярны в машинном переводе и суммаризации: $P(y_t|y_{1t-1}, x_{1:\text{len}(x)})$.

Модель PACSUM. Для кодировки предложений и вычисления матрицы сходств используется BERT (англ. *Bidirectional Encoder Representations from Transformers* – языковая модель, основанная на архитектуре «трансформер»). Каждое предложение в тексте кодируется единым вектором. Для дообучения представления предложений применяется *negative sampling* с использованием 2-х моделей BERT. Положительными примерами в данном случае являются фразы, смежные друг с другом, отрицательным – несмежные друг с другом фразы [5]. Фразы, которые должны стоять близко друг к другу, должны иметь схожий вектор.

Граф в этой модели является ориентированным (между двумя вершинами 2 ребра, которые имеют разные веса) $\lambda_1 e_{ij}$ и $\lambda_2 e_{ij}$, где e_{ij} – сходство между векторами предложений. Степень вершины вычисляется по формуле

$$centrality(s_i) = \lambda_1 \sum_{j<i} e_{ij} + \lambda_2 \sum_{j>i} e_{ij}.$$

Набор моделей BERTSumExt. В таких моделях в предобученный BERT подставляется текст, где перед каждым предложением ставится токен [CLS]. Эмбеддинги сегментов чередуются для разделения предложений. Далее представления для CLS токенов вытаскиваются с последнего слоя и идут в дополнительный трансформер, который уже используется для тэггинга (отбора предложений в аннотацию). Весь стек дообучается под конкретную задачу. Сравнение моделей оригинального BERT и BERT для суммаризации представлено на рис. 6.

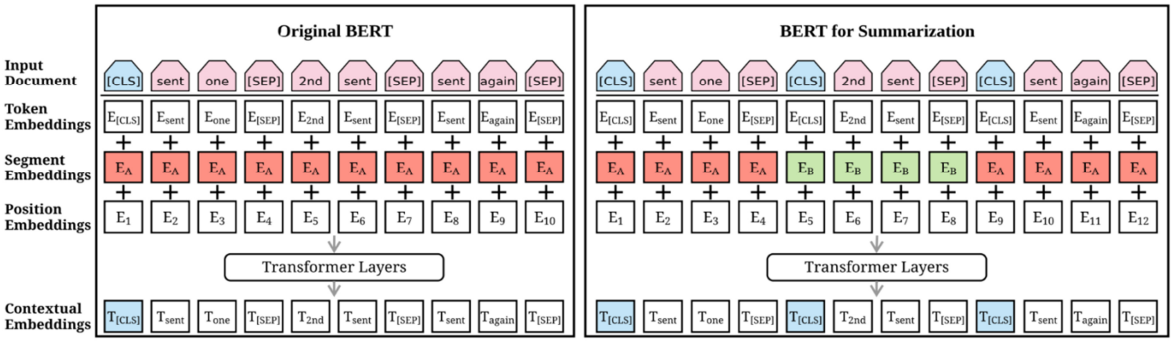

Рис. 6. Сравнение оригинальной модели BERT и усовершенствованной BERT для суммаризации

Fig. 6. Comparison of the original model BERT and modernized BERT for summarization

BERTSum является модификацией модели BERT, представленной в 2018 г. Модель BERT на сегодняшний день является эталоном для решения задач в области NLP, однако она не очень подходит к задаче автореферирования. Обычная BERT учится решать сразу 2 задачи: задачу маскированного языкового моделирования (*maskedlanguage model*) и задачу: является ли одно предложение продолжением другого (*next sentence prediction*). При таком подходе выходные векторы будут основаны на токенах, а нам нужны векторы, основанные на предложениях, поэтому для решения задачи нужно модифицировать входную последовательность. Как показано на рис. 6, токен CLS вставлен перед каждым предложением, а токен SEP – после каждого предложения. В оригинальном BERT CLS используется как символ для агрегации признаков из одного или пары предложений, в модифицированном же он используется для извлечения признаков из каждого предложения.

После получения векторов предложений строится несколько слоев, специально используемых для реферирования. Слои строятся поверх выходов BERT, чтобы получать на вход признаки на уровне документов. Для каждого предложения вычисляется финальное значение *Y*. Функцией потерь для всей модели будет бинарная кросс-энтропия [6]. Слои для суммаризации могут быть нескольких видов. Во-первых, это может быть обычный линейный слой, который получает ответ по формуле

$$\hat{Y} = \sigma(W_0 T_i + b_0),$$

где σ является сигмоидной функцией активации; W_0 – входы слоя; T_i – вектор для преобразования из верхнего слоя трансформатора; b_0 – смещение. Во-вторых, это может быть слой трансформера, ориентированный на задачу суммаризации из выходов BERT:

$$h = LN\left(h^{l-1} + MHAtt\left(h^{l-1}\right)\right),$$

где h^0 равен позиции вектора *T*, который, в свою очередь, является выходом BERT. Под *LN* понимается линейная нормализация, а под *MHAtt* – «*multi head attention*». Финальный слой – все та же сигмоидная функция:

$$\hat{Y} = \sigma(W_0 h_i^L + b_0),$$

где h_i^L – вектор предложения с финального слоя трансформера.

Абстрактивные подходы

Абстрактивный подход представляет собой генерацию нового текста, резюмирующего исходный. В таком резюме могут встречаться фразы, которые релевантны исходному тексту, но сами в нем не встречаются. Схема абстрактивной суммаризации представлена на рис. 7.

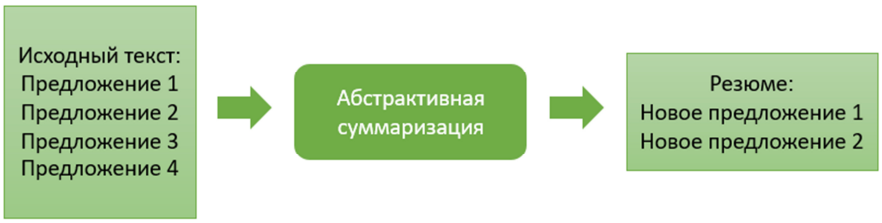

Рис. 7. Схема абстрактивного подхода суммаризации

Fig. 7. Graph of abstractive approach to summarization

Часто в алгоритмах абстрактивного подхода используется архитектура, представленная на рис. 8.

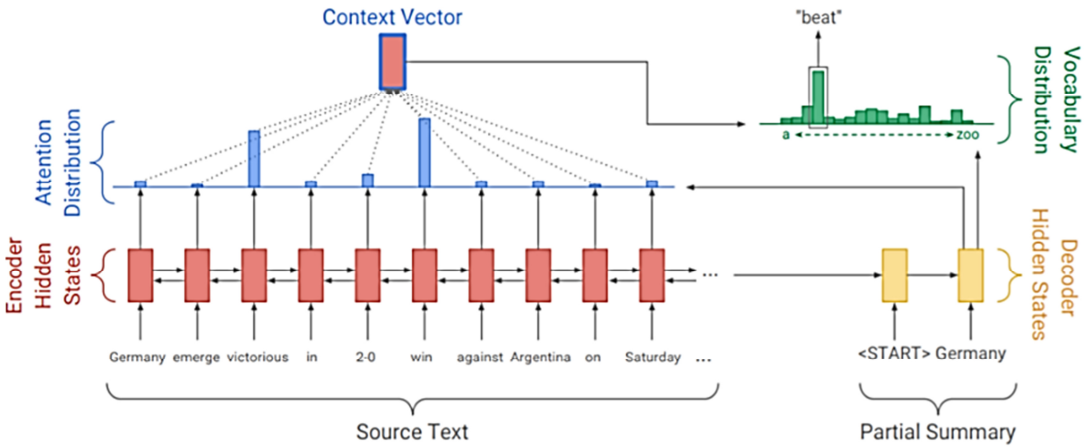

Рис. 8. Архитектура работы абстрактивного подхода

Fig. 8. Architecture of the abstractive approach

Encoder кодирует каждое слово контекстно-зависимым битингом. *Decoder* принимает на каждом шаге текущие слова в аннотации (которые уже известны) и пытается сгенерировать распределение вероятностей над следующим словом.

Вычисление весов внимания включает следующие этапы: сложение произведения матрицы W_h на векторы h_i, которые выдал кодировщик, произведения матрицы W_s на вектор состояния s_t и векторы кодировщика. После нелинейной активации получаются скалярные баллы (e – скалярный балл вектора i; b – обучаемый параметр):

$$e_i^t = v^T \tanh\left(W_h h_i + W_s s_t + b_{attn}\right).$$

Скалярные баллы нормализуются с помощью функции *softmax*:

$$a^t = softmax\left(e^t\right).$$

Взвешенная сумма полученных векторов представляет собой контекстный вектор, который будет использоваться вместе с текущим состоянием декодировщика при предсказании следующего слова:

$$h_t^* = \sum_i a_i^t h_i,$$

где h_i – представления векторов с *Encoder*'а, a_i^t – скалярные баллы.

Сеть Pointer Generator Network. Сеть Pointer Generator Network представляет собой модификацию типичной Seq2Seq архитектуры. Эта сеть может не только давать распределения из словаря, но и копировать слова из исходного текста. На каждом шаге генерации принимается решение: сгенерировать слово из словаря или скопировать слово из исходного текста. На основе контекстного вектора вычисляется вероятность, что следует генерировать слово:

$$p_{gen} = \sigma\left(w_h^T h_t^* + w_s^T s_t + w_x^T x_t + b_{ptr}\right).$$

При генерации слова вероятность какого-либо слова определяется суммой произведения вероятности генерации слова на вероятность словаря

и произведения вероятности копирования на сумму весов внимания, которые уделялись данному слову на текущей итерации:

$$P(w) = p_{gen} P_{vocab}(w) + (1 - p_{gen}) \sum_{i:w_i=w} a_i^t.$$

Для борьбы с повторениями в генерируемом тексте используется механизм покрытия (*coverage*). Он отслеживает сумму всех весов внимания, которые были за предыдущие шаги. Перед вычислением весов внимания на следующем шаге учитывается ранее уделенное внимание данному слову:

$$c^t = \sum_{t'=0}^{t-1} a^{t'}.$$

Если какое-то слово покрыто очень часто, то для него вычисляется штраф:

$$loss_t = -\log P(w_t^*) + \lambda \sum_i \min(a_i^t, c_i^t).$$

Схема сети Pointer Generator Network представлена на рис. 9 [7].

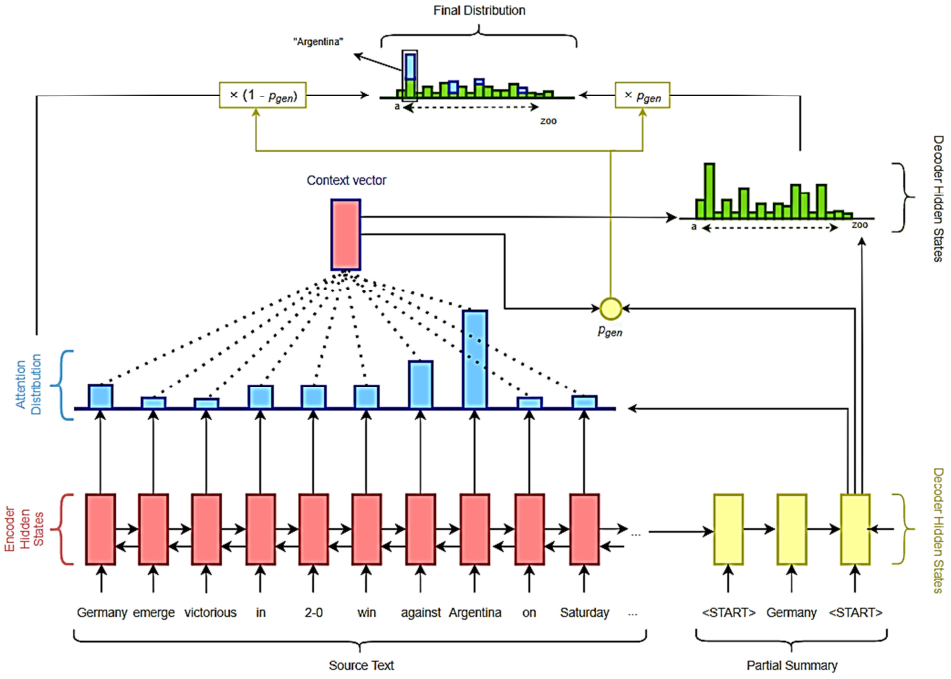

Рис. 9. Архитектура работы сети Pointer Generator Network

Fig. 9. Architecture of Pointer Generator Network

Сеть MBART. Сеть представляет собой реферирование при помощи предобученных трансформеров. Модель предобучается на восстановлении корректного входа из зашумленного.

Целью обучения является задача реконструкции текста. Перед подачей данных в модель часть токенов маскируется (заменяется на токен [MASK]), часть удаляется. В основе этой модели лежит концепция Seq2Seq, в качестве энкодера выступает модель BERT, а качестве декодера – GPT-2. Архитектура работы модели представлена на рис. 10.

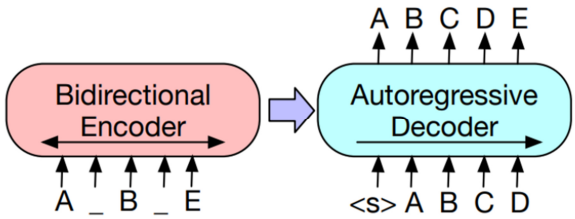

Рис. 10. Схема сети MBART

Fig. 10. Graph of MBART network

Случайные токены заменяются масками, документ кодируется двунаправленно. Отсутствующие токены предсказываются независимо. Они определяются автоматически, что означает, что GPT можно использовать для генерации.

Виды шумов:

1. Замена токенов на [MASK] (BERT objective).
2. Удаление токенов.
3. Замена последовательностей токенов на один токен [MASK], включая пустые последовательности.
4. Перемешивание предложений.
5. Ротация текста (текст начинается с рандомной позиции, а предыдущие позиции улетают в конец).

Графические схемы шумов представлены на рис. 11.

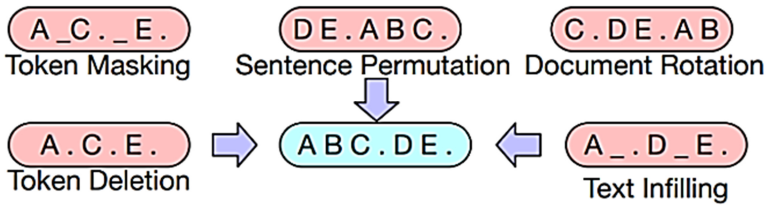

Рис. 11. Схемы шумов в модели MBART

Fig. 11. Graph of noises in the MBART model

Тюнинг модели MBART:

1. Классификация текстов – полный прогон через *Encoder* и *Decoder*, классификация последнего токена в декодере (рис. 12).

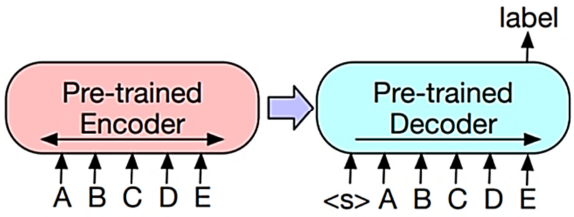

Рис. 12. Схема классификации текстов

Fig. 12. Graph of the text classification

2. Тэггинг – аналогично классификации, но классифицируются все токены в декодере (рис. 13).

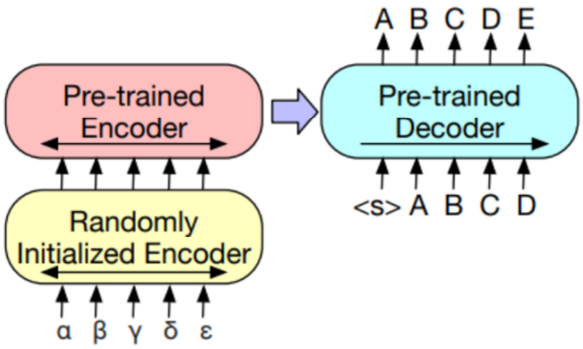

Рис. 13. Схема тэггинга текстов

Fig. 13. Graph of text tagging

3. Реферирование и Seq2Seq задачи для одного языка – *straightforward.*

Комбинированные подходы

Комбинированные подходы реферирования, по сравнению с абстрактивными методами, проще

в разработке, а по сравнению с экстрактивными методами – обеспечивают лучшее качество реферата. Сложность данных методов заключается в выборе удачного сочетания методик генерации и извлечения (абстрактивного и экстрактивного подходов). Примерами таких методов являются Scientific Text Summarizer [8].

Сравнительная характеристика подходов и моделей

Тексты статей, для которых формируются подводки, теоретически не имеют тематических ограничений. Экстрактивные методы автоматического реферирования могут извлекать предложения, слабо связанные между собой, но не требуют большого корпуса слов для обучения, и извлекающие алгоритмы проще в разработке. Абстрактивные модели создают более гладкие тексты. Генерирующие алгоритмы более сложные, но обрабатывают тексты не последовательно, а параллельно, что ускоряет обработку, и формируют реферативный текст со смысловыми вставками и подходящими связующими заменами. Однако данные модели требуют предобучения на больших корпусах новостных документов. Проанализировав различные подходы и модели реферирования текстов, можно сделать вывод, что, т. к. тексты в новостных статьях не специализированные, для формирования подводок более всего подходят абстрактивные модели Pointer General Network и MBART. Модель Pointer General Network наравне с генерацией новых токенов может копировать токены из исходной последовательности, расширяя итоговый словарь предсказания слов. При использовании второй из выбранных моделей можно использовать уже существующие веса, т. к. в процессе обучения она тренировалась на многих языках, включая русский. Данная модель хорошо подходит для автоматического формирования подводок определенных размеров для статей публикуемого издания на различных интернет-ресурсах.

Заключение

Ежедневно новостные порталы с целью повышения трафика сайта распространяют статьи в социальных сетях. Для повышения эффективности их работы используются NLP-методы суммаризации (автореферирования) текста. Рассмотрены основные подходы (абстрактивный и экстрактивный) к суммаризации, описаны их часто используемые модели. Проанализировав различные подходы и модели реферирования текстов, можно сделать вывод, что, т. к. тексты в новостных статьях не специализированные, для формирования подводок более всего подходят абстрактивные модели Pointer General Network и MBART.

Список источников

1. Шибайкин С. Д., Егунова А. И., Аббакумов А. А. Анализ применения нейронных сетей, градиентного бустинга и метода ближайших соседей для классификации нормативно-справочной информации // Науч.-техн. вестн. Поволжья. 2020. № 2. С. 54–58.

2. Федосин С. А., Плотникова Н. П., Немчинова Е. А., Денискин А. В. Применение адаптированного алгоритма Word2Vec в решении задач классификации и кластеризации нормативно-справочной информации // Науч.-техн. вестн. Поволжья. 2020. № 11. С. 120–126.

3. Федюшкин Н. А., Федосин С. А. О выборе методов векторизации текстовой информации // Науч.-техн. вестн. Поволжья. 2019. № 6. С. 129–134.

4. Афонин В. В., Никулин В. В. Оптимизация марковских систем массового обслуживания с отказами в системе MatLab // Вестн. Астрахан. гос. техн. ун-та. Сер.: Управление, вычислительная техника и информатика. 2018. № 1. С. 112–120.

5. Buyukkokten O., Garcia-Molina H., Paepcke A. Seeing the whole in parts: text summarization for web browsing on handheld devices // Proceedings of the 10th International Conference on World Wide Web. 2001. С. 652–662.

6. Luhn H. P. The Automatic Creation of Literature Abstracts // IBM Journal of Research and Development. 1958. V. 2. N. 2. P. 159–165. DOI: 10.1147 / rd.22.0159.

7. Linke-Ellis N. Closed captioning in America: Looking beyond compliance // Proceedings of the TAO Workshop on TV Closed Captions for the Hearing Impaired People. Tokyo, Japan. 1999. P. 43–59.

8. Батура Т. В., Бакиева А. М. Гибридный метод автореферирования научно-технических текстов на основе риторического анализа // Программные продукты и системы. 2020. Т. 33. № 1. С. 144–153. DOI: 10.15827/0236-235X.129.144-153.

References

1. Shibaikin S. D., Egunova A. I., Abbakumov A. A. Analiz primeneniia neironnykh setei, gradientnogo bustinga i metoda blizhaishikh sosedei dlia klassifikatsii normativno-spravochnoi informatsii [Analysis of using neural networks, gradient boosting and nearest neighbor method for classification of reference information]. *Nauchno-tekhnicheskii vestnik Povolzh'ia*, 2020, no. 2, pp. 54-58.

2. Fedosin S. A., Plotnikova N. P., Nemchinova E. A., Deniskin A. V. Primenenie adaptirovannogo algoritma Word2Vec v reshenii zadach klassifikatsii i klasterizatsii normativno-spravochnoi informatsii [Application of adapted word2vec algorithm in solving problems of classification and clustering of reference information]. *Nauchno-tekhnicheskii vestnik Povolzh'ia*, 2020, no. 11, pp. 120-126.

3. Fediushkin N. A., Fedosin S. A. O vybore metodov vektorizatsii tekstovoi informatsii [On choosing methods for vectorizing textual information]. *Nauchno-tekhnicheskii vestnik Povolzh'ia*, 2019, no. 6, pp. 129-134.

4. Afonin V. V., Nikulin V. V. Optimizatsiia markovskikh sistem massovogo obsluzhivaniia s otkazami v sisteme MatLab [Optimization of Markov queuing systems with failures in MatLab system]. *Vestnik Astrakhanskogo gosudarstvennogo*

tekhnicheskogo universiteta. Seriia: Upravlenie, vychislitel'naia tekhnika i informatika, 2018, no. 1, pp. 112-120.

5. Buyukkokten O., Garcia-Molina H., Paepcke A. Seeing the whole in parts: text summarization for web browsing on handheld devices. *Proceedings of the 10th International Conference on World Wide Web*, 2001, pp. 652-662.

6. Luhn H. P. The Automatic Creation of Literature Abstracts. *IBM Journal of Research and Development*, 1958, vol. 2, no. 2, pp. 159-165. DOI: 10.1147 / rd.22.0159.

7. Linke-Ellis N. Closed captioning in America: Looking beyond compliance. *Proceedings of the TAO Workshop on TV Closed Captions for the Hearing Impaired People*. Tokyo, Japan, 1999. Pp. 43-59.

8. Batura T. V., Bakiyeva A. M. A hybrid method for automatic summarization of scientific and technical texts based on rhetorical analysis. *Software & Systems*. 2020, vol. 33, no. 1, pp. 144–153 (in Russ.). DOI: 10.15827/0236-235X.129.144-153.

Статья поступила в редакцию 19.09.2022; одобрена после рецензирования 08.12.2022; принята к публикации 12.01.2023
The article is submitted 19.09.2022; approved after reviewing 08.12.2022; accepted for publication 12.01.2023

Информация об авторах / Information about the authors

Алла Ивановна Егунова – кандидат исторических наук, доцент; доцент кафедры автоматизированных систем обработки информации и управления; Национальный исследовательский Мордовский государственный университет им. Н. П. Огарева; vystasalla@gmail.com

Роман Сергеевич Комаров – магистрант кафедры автоматизированных систем обработки информации и управления; Национальный исследовательский Мордовский государственный университет им. Н. П. Огарева; roman071999@mail.ru

Юлия Сергеевна Вечканова – аспирант кафедры автоматизированных систем обработки информации и управления; Национальный исследовательский Мордовский государственный университет им. Н. П. Огарева; yuliya_kolushova@mail.ru

Ольга Игоревна Егунова – студент кафедры автоматизированных систем обработки информации и управления; Национальный исследовательский Мордовский государственный университет им. Н. П. Огарева; Olga-egunova00@rambler.ru

Дмитрий Петрович Сидоров – кандидат технических наук, доцент; доцент кафедры автоматизированных систем обработки информации и управления; Национальный исследовательский Мордовский государственный университет им. Н. П. Огарева; sidorovd@mail.ru

Сергей Дмитриевич Шибайкин – кандидат технических наук; доцент кафедры инфокоммуникационных технологий и систем связи; Национальный исследовательский Мордовский государственный университет им. Н. П. Огарева; shibaikinsd@rambler.ru

Владимир Валерьевич Никулин – кандидат технических наук, доцент; заведующий кафедрой инфокоммуникационных технологий и систем связи; Национальный исследовательский Мордовский государственный университет им. Н. П. Огарева; nikulinvv@mail.ru

Alla I. Egunova – Candidate of Sciences in History, Assistant Professor; Assistant Professor of the Department of Automated Information Processing and Control Systems; National Research Ogarev Mordovia State University; vystasalla@gmail.com

Roman S. Komarov – Master's Course Student of the Department of Automated Information Processing and Control Systems; National Research Ogarev Mordovia State University; roman071999@mail.ru

Yuliya S. Vechkanova – Postgraduate Student of the Department of Automated Information Processing and Control Systems; National Research Ogarev Mordovia State University; yuliya_kolushova@mail.ru

Olga I. Egunova – Student of the Department of Automated Information Processing and Control Systems; National Research Ogarev Mordovia State University; egunova00@rambler.ru

Dmitry P. Sidorov – Candidate of Sciences in Technology, Assistant Professor; Assistant Professor of the Department of Automated Information Processing and Control Systems; National Research Ogarev Mordovia State University; sidorovd@mail.ru

Sergei D. Shibaikin – Candidate of Sciences in Technology; Assistant Professor of the Department of Infocommunication Technologies and Communication Systems; National Research Ogarev Mordovia State University; shibaikinsd@rambler.ru

Vladimir V. Nikulin – Candidate of Technical Sciences, Assistant Professor; Head of the Department of Infocommunication Technologies and Communication Systems; National Research Ogarev Mordovia State University; nikulinvv@mail.ru