УДК 004.9 DOI: 10.18698/2541-8009-2021-02-673

ГЕНЕРАЦИЯ ТЕКСТОВЫХ КОНТЕЙНЕРОВ С ИСПОЛЬЗОВАНИЕМ АППАРАТА НЕЙРОННЫХ СЕТЕЙ

Н.В. Назаренко nazarenkonv@student.bmstu.ru
SPIN-код: 5584-8020

Д.Е. Бекасов bekasov@bmstu.ru
SPIN-код: 6513-4597

МГТУ им. Н.Э. Баумана, Москва, Российская Федерация

Аннотация

Предложен вариант реализации метода генерации текстовых контейнеров с использованием аппарата нейронных сетей и языковой модели, которая была обучена на корпусе русскоязычных текстов и затем дообучена на корпусе классической литературы. Для разработанного метода было выполнено сравнение результатов работы разных алгоритмов на этапе кодирования исходного встраиваемого сообщения по ряду параметров и выбран наилучший из них для возможного использования в задачах интеграции данных и последующей передачи значащей информации по доступным каналам связи. На основании анализа полученных результатов работы разработанного метода были предложены возможные направления его дальнейшего развития.

Ключевые слова

Генеративные нейронные сети, GPT-2, токен, токенизатор, языковая модель, контекст, текст, контейнер, корпус

Поступила в редакцию 16.12.2020

Введение. В настоящее время с целью обеспечения конфиденциальности и аутентичности информации активно применяются методы передачи значащей информации путем сокрытия содержания сообщения. Однако основная проблема подобных методов заключается в том, что они не позволяют скрыть сам факт передачи информации. Для решения данной проблемы применяется стеганография — способ передачи сообщения в каком-либо контейнере (тексте, изображении, звуке и т. д.), который сохраняет в тайне сам факт наличия каких-либо сообщений [1].

Таким образом, текстовые контейнеры используются в задаче лингвистической стеганографии [2], которая и рассматривается в данной работе. Все множество методов лингвистической стеганографии можно подразделить на две группы: генеративные методы и методы, основанные на редактировании. Примером метода из второй группы может быть метод подстановки синонимов: в этом случае содержимое сопроводительного текста для контейнера выбирается человеком и слегка модифицируется с помощью словаря синонимов для кодирования информации. Данный алгоритм достаточно сложен для реализации в ма-

шинном виде, особенно из-за видоизменения слов. В рамках данной работы интерес представляют именно генеративные методы, основанные на использовании аппарата нейронных сетей. Они позволяют сгенерировать полностью новый и оригинальный текст.

При описании методов лингвистической стеганографии применяется следующий понятийный аппарат:

– стеготекст (текст на естественном языке, содержащий значимую информацию);

– контейнер (любая информация, используемая для сокрытия значимого сообщения);

– контекст (базовый текстовый контейнер);

– токен (единица естественного языка (слово или его часть), представляемая в виде вектора чисел);

– поток токенов (непрерывный подряд идущий набор токенов);

– языковая модель (математическая модель представления естественного языка в виде векторов; в данной работе это нейронная сеть, обученная на большом количестве текстов).

Генеративная лингвистическая стеганография. Обобщенный алгоритм работы генеративного метода лингвистической стеганографии имеет следующий вид [3].

1. Отправитель генерирует стеготекст:

$$y = f(S, P_l),$$

где f — некоторая обратимая функция; S — скрываемые данные, P_l – языковая модель.

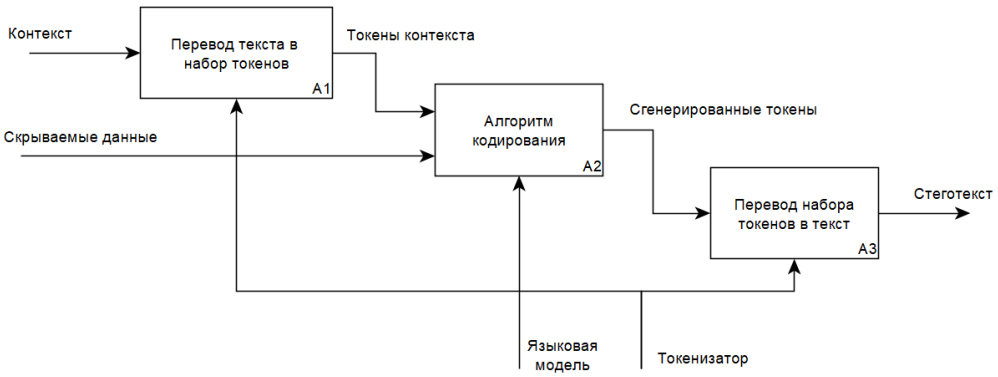

Схема процесса генерации стеготекста

2. Стеготекст y передается по открытому каналу.

3. Получатель принимает стеготекст и извлекает скрытые данные:

$$S = f^{-1}(y, P_l).$$

Общий процесс генерации стеготекста изображен на рисунке. Для обучения любой языковой модели требуется разбить входные данные, на которых обучается модель, на некоторые токены. Соответственно, в дальнейшем языковая модель будет работать только с этими токенами. В генеративных методах используются N предыдущих токенов для выбора того, какой токен будет сгенерирован следующим на выходе. Поэтому для того, чтобы генерировались более естественные тексты, используется инициализирующий контекст. Данный инициализирующий контекст перед началом работы с помощью токенизатора переводится в поток токенов, принимаемых обученной языковой моделью. Для того чтобы токенизатор разбивал исходный текст на токены, которые понимает языковая модель, его необходимо обучать вместе с этой моделью.

Все существующие на данный момент генеративные методы лингвистической стеганографии очень похожи между собой. Они используют языковые модели, представляющие собой, как правило, заранее обученные на большом корпусе текстов нейронные сети [3–6]. Одно из главных отличий таких методов — это алгоритм подбора следующего токена по скрываемой информации.

В данной работе в качестве языковой модели используется архитектура нейронной сети GPT-2 [7], обученная предсказывать следующее слово в предложении на естественном языке. Изначально в качестве архитектур для языковых моделей использовались рекуррентные нейронные сети (RNN), в частности, LSTM [3]. Но после разработки в 2017 г. архитектуры Transformer [8] она стала постепенно преобладать в задачах генерации текстов. Несмотря на то что у оригинального Transformer имеется проблема с запоминанием длинных последовательностей (LSTM может запоминать более длинные), быстрота его обучения и глубина сети компенсировали это [9].

После определения языковой модели необходимо выбрать алгоритмы кодирования исходной скрываемой информации, по которым затем будут выбираться токены генерируемого текста. В процессе анализа было найдено три возможных алгоритма: кодирование на основе блоков, кодирование с помощью дерева Хаффмана и арифметическое кодирование. Ниже эти три алгоритма рассматриваются и сравниваются между собой более подробно.

Кодирование на основе блоков. Ниже представлен алгоритм метода, основанного на разбиении информации на блоки.

1. Отправитель и получатель договариваются об общем ключе, который представляет собой словарь токенов.

2. Скрываемые данные S разделяются на битовые блоки длиной $\left|B\right|$.

3. Выбранный ранее словарь случайным образом разбивается на $2^{|B|}$ непересекающихся наборов токенов (бинов), случайно выбранных из словаря. Каждый токен появляется ровно в одном бине, и каждый бин содержит $\left|V\right| / 2^{|B|}$ токенов и индексируется битовым блоком длиной $\left|B\right|$.

4. Для каждого битового блока B, полученного в п. 1 с помощью языковой модели, выбирается один токен из бина W_B.

В работе [3] в качестве языковой модели использовалась двухслойная сеть LSTM, обученная на постах Twitter (твитах). При использовании блочного подхода с $|B| = 4$ и описанной языковой моделью в одном твите (содержащем в среднем 16,04 слов) удалось встроить 32 бита полезной информации.

Кодирование с помощью дерева Хаффмана. Метод основан на построении дерева Хаффмана на каждом шаге t на основе языковой модели и кодировании данных в битовых блоках через дерево Хаффмана [5].

В методе используется словарь V размера $|V|$, который является конечным набором токенов. Также применяется расширенный словарь V^* — множество всех конечных последовательностей токенов из V (т. е. всех текстов). Алгоритм одного шага кодирования представлен ниже. Под префиксом в данном алгоритме понимается уже существующая часть текста — первое инициализирующее слово или набор слов, сгенерированных на предыдущих шагах. Сам алгоритм имеет следующий вид.

1. Вычисление распределения p следующего токена с помощью языковой модели для текущего префикса.

2. Построение дерева Хаффмана c для p.

3. Кодирование битового блока скрываемых данных S с помощью дерева Хаффмана c, которое можно описать следующим образом. Берется очередной бит скрываемого сообщения S. Если бит равен нулю, то выбирается левое поддерево, если бит равен единице, то правое. Процесс продолжается до тех пор, пока не будет достигнут лист (узел, не имеющий потомков).

4. Полученный токен добавляется к префиксу.

Арифметическое кодирование. Метод основан на построении интервалов кодирования на каждом шаге t на основе языковой модели и сокрытия полезных данных с помощью арифметического кодирования [6]. Чтобы закодировать в текст скрываемые данные, эти данные S рассматриваются как двоичное представление дроби в диапазоне $[0, 1)$. Ниже приведено описание одного шага алгоритма.

1. Вычисление распределения p следующего токена с помощью языковой модели для текущего префикса.

2. Построение интервалов арифметического кодирования для распределения p.

3. Выбор следующего токена на основе того, какому интервалу принадлежит дробное представление скрываемого сообщения.

4. Добавление полученного токена к префиксу.

Метрики качества алгоритмов лингвистической стеганографии. В одной из последних работ на тему лингвистической стеганографии [6], для сравнения генеративных методов предлагается использовать расстояние Кульбака — Лейблера (D_{KL}). Это расстояние является мерой удаленности друг от друга двух вероятностных распределений:

$$D_{KL}(q \,||\, P_{true}),$$

где P_{true} — истинное распределение естественного языка, q — вероятностное распределение сгенерированного текстового контейнера.

Если $D_{KL} = 0$, то система считается полностью защищенной, поскольку потенциальный перехватчик не сможет отличить закодированные сообщения от сообщений, написанных человеком. Таким образом, с точки зрения теории информации, цель безопасности лингвистической стеганографии состоит в том, чтобы минимизировать D_{KL}.

Помимо D_{KL} важным критерием для сравнения методов является отношение «биты/слово» (*B*/*W*), которое показывает, сколько битов данных можно встроить в одно слово сгенерированного текста.

Также в работе [3] для сравнения методов используется понятие «растерянность» (*PPL*, англ. *perplexity*) — измерение того, насколько хорошо распределение вероятностей или модель вероятности предсказывает выборку. Данный параметр может использоваться для сравнения вероятностных моделей. Низкая растерянность указывает на то, что распределение вероятностей хорошо подходит для прогнозирования выборки.

Для расчета растерянности используют формулу

$$PPL = \exp\left(-\frac{1}{N}\sum_{x} \ln p(x)\right).$$

Сравнение генеративных лингвистических методов. Необходимо произвести сравнение описанных выше методов кодирования для выбора наилучшего из них по указанным метрикам качества, чтобы впоследствии можно было определить какой метод стоит использовать при генерации стеготекста. Для сравнения методов в данной работе использовалась языковая модель архитектуры GPT-2, имеющая 355 миллионов параметров. Данная языковая модель была обучена на большом корпусе (~230 Гб) русскоязычных текстов и дообучена на корпусе классической литературы (~500 Мб) [10].

Описанные выше метрики для кодирования на основе блоков при изменении параметра длины битовых блоков представлены в табл. 1.

Таблица 1

Результаты сравнения для блочного подхода

Длина битового окна	D_{KL}	*B*/*W*	*PPL*
1	1,5797	1,53	20,8483
2	1,4516	3,02	99,8653
3	0,9412	3,95	176,6577
4	0,4239	5,40	204,8231

При построении дерева Хаффмана на каждом шаге из распределения p выбирается N токенов с наибольшей вероятностью. При изменении данного параметра были получены результаты, представленные в табл. 2.

Таблица 2

Результаты сравнения для дерева Хаффмана

N	D_{KL}	B/W	PPL
2	0,7592	1,24	9,0270
4	0,2733	2,11	12,4293
8	0,0418	3,18	15,9975
16	0,0713	4,02	21,4434

Общие сводные результаты исследования всех трех методов кодирования приведены в табл. 3. Как можно заметить, наименьшее D_{KL} имеет метод кодирования с использованием дерева Хаффмана, однако при этом он позволяет скрыть наименьшее число бит в одном слове и имеет среднее значение растерянности. Поэтому среди данных методов лучше всего себя показало арифметическое кодирование. Оно имеет наименьшую растерянность и мало отличается по остальным параметрам по сравнению с другими методами.

Таблица 3

Результаты сравнения методов кодирования

Метод	$\min D_{kl}$	B/W	PPL
Блочный подход ($\|B\| = 4$)	0,4239	5,40	204,8231
Дерево Хаффмана ($N = 8$)	0,0418	3,18	15,9975
Арифметическое кодирование	0,0651	5,09	3,2779

Пример работы. Пример работы метода лингвистической стеганографии с заданным контекстом и скрываемым сообщением приведен ниже:

Контекст. Он в довольном расположении духа сидел в своей бричке, катившейся давно по столбовой дороге. Из предыдущей главы уже видно, в чем состоял главный предмет его вкуса и склонностей, а потому не диво, что он скоро погрузился весь в него и телом и душою. Предположения, сметы и соображения, блуждавшие по лицу его, видно, были очень приятны, ибо ежеминутно оставляли после себя следы довольной усмешки.

Скрываемое сообщение. Важное сообщение.

Стеготекст. Видно, ему было весело. Он прислонился к уголку, подставив кулак светловолосой голове своей, и смотрел с невольным любованием на проносящиеся мимо и мелькавшие вдали. Ветер был хорош: заливал глаза, ласкал щеки и груди; ночи были недальние, небо чистое, звезд не видно было.

Результатом работы метода является сгенерированный стеготекст, в котором содержится само сообщение и который является продолжением контекста.

Возможные направления дальнейшего развития метода. Как видно из предыдущего примера, сгенерированный стеготекст может оказаться не законченным, т. е. прерванным посреди предложения, из-за чего теряется общая ло-

гичность текста. Таким образом, в качестве одного из направлений дальнейшего развития можно поставить задачу дополнения стеготекста до законченного предложения для повышения его естественности.

Помимо этого, как было описано раннее, прежде чем исходный текст буден подан на вход языковой модели, он должен быть разбит на токены с помощью токенизатора. Аналогично, на выходе из языковой модели новый текст строится из сгенерированных ею токенов.

Для того чтобы декодировать закодированное сообщение, необходимо на вход языковой модели-декодера, помимо исходного контекста, подать сам сгенерированный текст с этим закодированным сообщением, который также должен быть предварительно разбит на токены. Однако разбиение на токены токенизатором на входе декодера и разбиение на токены при генерации текста во время кодирования может оказаться разным. В итоге восстановленное при дешифрации сообщение может оказаться частично или полностью поврежденным и некорректным. Следовательно, необходимо предпринять меры для исключения данной ситуации.

Для этого предлагается в дальнейшем использовать разбиение исходного скрываемого сообщения на пакеты — последовательности с фиксированным количеством битов с контрольной суммой в конце. Тем самым, зная, где находится начало пакета, а где конец, и его контрольную сумму можно подобрать такое разбиение на токены на входе декодера, которое было при генерации текста языковой моделью. Более того, с использованием контрольной суммы можно узнать, не были ли повреждены или изменены данные при пересылке всего сообщения.

Заключение. В данной работе рассмотрен вариант реализации генеративного метода лингвистической стеганографии на основе использования различных способов кодирования встраиваемой значимой информации и произведено сравнение этих способов с использованием описанных метрик качества. В ходе сравнения минимальное значение D_{KL} (0,0418) получено с использованием дерева Хаффмана, однако наилучшим по остальным показателям оказалось арифметическое кодирование. Помимо этого сформулированы и предложены направления дальнейшего развития метода.

Литература

[1] Конахович Г.Ф., Пузыренко А.Ю. Компьютерная стеганография. Теория и практика. Киев, МК-Пресс, 2006.

[2] Бабина О.И. Лингвистическая стеганография: современные подходы. Ч. 1. *Вестник ЮУрГУ. Сер. Лингвистика,* 2015, т. 12, № 3, с. 27–33.

[3] Fang T., Jaggi M., Argyraki K. Generating steganographic text with LSTMs. *arxiv.org: веб-сайт.* URL: https://arxiv.org/abs/1705.10742 (дата обращения: 20.08.2020).

[4] Cox I., Kalker T., Pakura G., et al. Information transmission and steganography. In: Digital watermarking. Springer, 2005, pp. 15–29.

[5] Yang Z., Guo X., Chen Z., et al. RNN-Stega: linguistic steganography based on recurrent neural networks. *IEEE Trans. Inf. Forensics Security*, 2019, vol. 14, no. 5, pp. 1280–1295. DOI: https://doi.org/10.1109/TIFS.2018.2871746

[6] Ziegler Z., Deng Y. Neural linguistic steganography. *Proc. EMNLP-IJCNLP*, 2019, pp. 1210–1215. DOI: http://dx.doi.org/10.18653/v1/D19-1115
[7] Radford A., Wu J., Child R., et al. Language models are unsupervised multitask learners. URL: https://aisc.ai.science/static/slides/20190307_EhsanAmjadian.pdf (дата обращения: 20.08.2020).
[8] Vaswani A., Shazeer N., Parmar N., et al. Attention is all you need. *Proc. NIPS*, 2017. URL: https://dl.acm.org/doi/10.5555/3295222.3295349 (дата обращения: 20.08.2020).
[9] Greene M. LSTM vs transformer within semantic parsing. *yale-lily.github.io: веб-сайт.* URL: https://yale-lily.github.io/public/matt_f2018.pdf (дата обращения: 20.08.2020).
[10] Grankin M. Russian GPT-2. *github.com: веб-сайт.* URL: https://github.com/ mgrankin/ru_transformers (дата обращения: 20.08.2020).

Назаренко Никита Вадимович — студент кафедры «Программное обеспечение ЭВМ и информационные технологии», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

Бекасов Денис Евгеньевич — старший преподаватель кафедры «Программное обеспечение ЭВМ и информационные технологии», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

Научный руководитель — Барышникова Марина Юрьевна, кандидат педагогических наук, доцент кафедры «Программное обеспечение ЭВМ и информационные технологии», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

Ссылку на эту статью просим оформлять следующим образом:
Назаренко Н.В., Бекасов Д.Е. Генерация текстовых контейнеров с использованием аппарата нейронных сетей. *Политехнический молодежный журнал*, 2021, № 02(55). http://dx.doi.org/ 10.18698/2541-8009-2021-02-673

GENERATION OF TEXT CONTAINERS USING THE NEURAL NETWORK APPARATUS

N.V. Nazarenko nazarenkonv@student.bmstu.ru
SPIN-code: 5584-8020

D.E. Bekasov bekasov@bmstu.ru
SPIN-code: 6513-4597

Bauman Moscow State Technical University, Moscow, Russian Federation

Abstract

The paper describes an implementation option of the method for generating text containers using the apparatus of neural networks and a language model, which was trained on the Russian-language texts corpus and then retrained on the classical literature corpus. For the developed method, the results of the operation of different algorithms were compared at the stage of encoding the original embedded message for a number of parameters and the best algorithm was selected for possible use in data integration problems and subsequent transmission of meaningful information through available communication channels. Based on the analysis of the developed method results, possible directions for its further development were proposed.

Keywords

Generative neural networks, GPT-2, token, tokenizer, language model, context, text, container, corpus

Received 16.12.2020

References

[1] Konakhovich G.F., Puzyrenko A.Yu. Komp'yuternaya steganografiya. Teoriya i praktika [Computer stenography. Theory and practice]. Kiev, MK-Press Publ., 2006 (in Russ.).

[2] Babina O.I. Linguistic steganography: state-of-the-art. Part 1. *Vestnik YuUrGU. Ser. Lingvistika* [Bulletin of the South Ural State University. Ser. Linguistics], 2015, vol. 12, no. 3, pp. 27–33 (in Russ.).

[3] Fang T., Jaggi M., Argyraki K. Generating steganographic text with LSTMs. *arxiv.org: website*. URL: https://arxiv.org/abs/1705.10742 (accessed: 20.08.2020).

[4] Cox I., Kalker T., Pakura G., et al. Information transmission and steganography. In: Digital Watermarking. Springer, 2005, pp. 15–29.

[5] Yang Z., Guo X., Chen Z., et al. RNN-Stega: linguistic steganography based on recurrent neural networks. *IEEE Trans. Inf. Forensics Security*, 2019, vol. 14, no. 5, pp. 1280–1295. DOI: https://doi.org/10.1109/TIFS.2018.2871746

[6] Ziegler Z., Deng Y. Neural linguistic steganography. *Proc. EMNLP-IJCNLP*, 2019, pp. 1210–1215. DOI: http://dx.doi.org/10.18653/v1/D19-1115

[7] Radford A., Wu J., Child R., et al. Language models are unsupervised multitask learners. URL: https://aisc.ai.science/static/slides/20190307_EhsanAmjadian.pdf (accessed: 20.08.2020).

[8] Vaswani A., Shazeer N., Parmar N., et al. Attention is all you need. *Proc. NIPS*, 2017. URL: https://dl.acm.org/doi/10.5555/3295222.3295349 (accessed: 20.08.2020).

[9] Greene M. LSTM vs transformer within semantic parsing. *yale-lily.github.io: website*. URL: https://yale-lily.github.io/public/matt_f2018.pdf (accessed: 20.08.2020).
[10] Grankin M. Russian GPT-2. *github.com: website*. URL: https://github.com/mgrankin/ru_transformers (accessed: 20.08.2020).

Nazarenko N.V. — Student, Department of Computer Software and Information Technologies, Bauman Moscow State Technical University, Moscow, Russian Federation.

Bekasov D.E. — Senior Lecturer, Department of Computer Software and Information Technologies, Bauman Moscow State Technical University, Moscow, Russian Federation.

Scientific advisor — Baryshnikova M.Yu., Cand. Sc. (Pedagog.), Assoc. Professor, Department of Computer Software and Information Technologies, Bauman Moscow State Technical University, Moscow, Russian Federation.

Please cite this article in English as:
Nazarenko N.V., Bekasov D.E. Generation of text containers using the neural network apparatus. *Politekhnicheskiy molodezhnyy zhurnal* [Politechnical student journal], 2021, no. 02(55). http://dx.doi.org/ 10.18698/2541-8009-2021-02-673.html (in Russ.).