|  |
| --- |
|  |
| ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА НА ТЕМУ:  **«Генерация сказок»** |
| 2022 год |

**Оглавление**

[​ Описание задачи 3](#__RefHeading___Toc127_2148401760)

[​ Наименование разработки 3](#__RefHeading___Toc129_2148401760)

[​ Цель и назначение разработки 3](#__RefHeading___Toc131_2148401760)

[​ Описание применения 3](#__RefHeading___Toc133_2148401760)

[​ Описание существующих решений 3](#__RefHeading___Toc135_2148401760)

[​ Описание данных 5](#__RefHeading___Toc137_2148401760)

[​ Исходные данные 5](#__RefHeading___Toc139_2148401760)

[​ Обработка данных для обучения 7](#__RefHeading___Toc141_2148401760)

[​ Архитектура решения 8](#__RefHeading___Toc143_2148401760)

[​ Описание модели 8](#__RefHeading___Toc145_2148401760)

[​ Описание процесса обучения 9](#__RefHeading___Toc147_2148401760)

[​ Описание процесса генерации 9](#__RefHeading___Toc1907_3871384179)

[​ Оценка качества решения 10](#__RefHeading___Toc424_784594672)

[​ Выводы и результаты работы 13](#__RefHeading___Toc426_784594672)

[​ Список литературы 15](#__RefHeading___Toc428_784594672)

[​ ПРИЛОЖЕНИЕ А. Сравнительная таблица с оценками качества 16](#__RefHeading___Toc1909_3871384179)

[​ ПРИЛОЖЕНИЕ Б. Примеры сгенерированных текстовых 18](#__RefHeading___Toc1919_3871384179)

# Описание задачи

## Наименование разработки

Наименование: "Модель генерации русских народных сказок" (далее — Модель)

Материалы разработки (проект) размещены в сети Интернет по адресу:

* https://github.com/edbons/faiky-tails.

## Цель и назначение разработки

Цель: снижение себестоимости контента, предоставляемого конечному потребителю, за счет использование уникальных произведений, не требующих отчислений правообладателям.

Назначение: создание уникальных текстовых и/или аудио произведений на тематику народных сказаний.

## Описание применения

Модель может применяться в качестве источника контента для следующих служб:

* голосовые помощники (пример, Алиса (Яндекс), Маруся (Mail.ru Group));
* службы потокового аудио (пример, Яндекс Музыка, Spotify).

Потребителями (пользователями) контента, генерируемого моделью, могут выступать:

* дети дошкольного возраста, для которых форма произведения превалирует над возможной несвязанностью содержания.
* родители детей, приобретающие подписки для служб потокового аудио.

## Описание существующих решений

Для генерации сказок в русскоязычном сегменте сети Интернет присутствуют различные ресурсы, среди которых для анализа были отобраны следующие:

1. Навык "Сказки" для голосового помощника Алиса (источник -- https://alice.voice-ui.ru/skills/ec1176f1-skazki)
2. Мобильное приложение "Сказаврия" (источник -- https://skazavria.ru/)
3. Сайт http://generatorskazok.ru/
4. Модель генерации сказок с озвучкой и картинками от Sber AI (источник -- https://github.com/sberbank-ai/DigiTeller/blob/main/Readme\_Rus.md)
5. Сервис "Балабоба" от Яндекс (источник -- <https://yandex.ru/lab/yalm?style=0>).
6. Сервис генерации прозы и стихов в стиле А. С. Пушкина <https://pushkin.monetka.name/>. Пользователь может задать собственные названия стиха или прозы, и «уровень креативности» (для генеративных моделей параметр температуры).

Для английского языка отмечены следующие ресурсы:

1. Сервис генерации историй <https://novelai.net/>. Имеет платные подписки. Для генерации использует дообученные модели семейства GPT, но какие конкретно установить до конца не получилось, т. к. проект закрытый. По некоторым сведениям это GPT-Neo (открытый аналог GPT-3) с вариантами вплоть до 6B параметров.
2. Сервис генерации текстов и картинок к ним <https://fairytailor.org/>. Носит исследовательский характер. Использует в качестве модели дообученную версию GPT-2 Medium, для генерации картинок используется модель CLIP.

Для генерации сказок в существующих решениях используются следующие подходы:

1. Подстановка в шаблон текста ключевых слов (фраз). Например, имена главных героев, сказочных артефактов. Подход используется в мобильном приложении "Сказаврия", на сайте http://generatorskazok.ru/.
2. Поиск существующей сказки по ключевым фразам пользователя. Не является генератором уникального контента, например навык "Сказки" для голосового помощника Алиса.
3. Генерация уникального текста большими моделями (пример, BERT, GPT, YAML) дообученными на специализированной доменной области (к примеру, на корпусе текстов художественной литературы), пример модели GPT-3 от Sber AI и сервис "Балабоба" от Яндекс.

Пример сгенерированного текста для фразы  "сказка про царя" в сервисе "Балабоба" от Яндекс:

*"сказка про царя, который в шутку женился на лягушке.*

*Только вот потом он обнаружил, что женился на крокодиле, и лягушачьи лапки оказались как раз кстати, чтобы заменить ему ужин.*

*А уж после того как царь обнаружил, что женится на утке, которая летает, а гнездо свила на вершине башни, он вообще чуть не ослеп и вынужден был бежать, потому как не смог отличить, где небо, а где земля."*

# Описание данных

## **Исходные данные**

В качестве источника текстов используются следующие ресурсы:

* <http://hyaenidae.narod.ru/>, на сайте представлены только русские народные сказки;
* <http://az.lib.ru/>, на сайте представлены литературные произведения различных жанров. При формировании данных для обучения с сайта были отобраны тексты жанра «Сказки».

Загрузка данных с сайта выполнена автоматизированным способом: скрипт web\_scraper.py.

В качестве исходного формата хранения необработанных текстов выбран CSV.

Анализ текстов представлен в файле: ./analyse\_texts.ipynb.

Общее число документов: 582

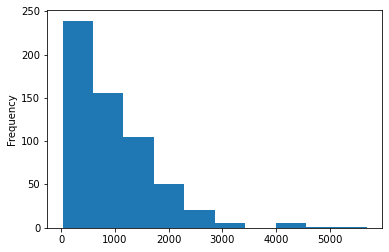
Статистика по словам в корпусе:

* Всего не уникальных слов: около 550 000;
* Среднее число слов в документе: около 950;
* Минимальное число слов в документе: около 26;
* Максимальное число слов в документе: около 5700.

Статистика по предложениям в корпусе:

* Среднее число предложений в документе: около 65;
* Минимальное число предложений в документе: 1;
* Максимальное число предложений в документе: около 400;
* Среднее число слов в предложении: около 10.

График распределения частоты слов в документах приведен на рисунке 1.

Рисунок 1 - График распределения частоты токенов в документах

Схематическое изображение наиболее часто используемых слов в документах («облако слов») приведено на рисунке 2.

Рисунок 2 - Наиболее часто используемые слова в документах («облако слов»)

Результаты и выводы анализа исходных данных:

1. Большинство произведений содержат до 1000 слов, что укладывается в размеры входных последовательностей современных языковых моделей (например, GPT-2 1024 токена, GPT-3 2048 токенов).
2. Для настоящего проекта в качестве максимальное длинны генерируемого текста определена длина 512 токенов, для удобства ручного анализа качества.

## Обработка данных для обучения

Для обучения универсальных языковых моделей (к примеру, GPT) на определенную доменную область (к примеру, тексты в стиле новостей, или художественных произведений) могут применяться следующие подходы:

1. Дообучение на специфичной выборке («fine tuning»). Предполагает небольшие изменения весов языковой модели при обучении на малом количестве данных. Требуются вычислительные ресурсы достаточные для цикла обучения.
2. Обучение на примере входной последовательности слов, методы Zero Shot, Few Shots. При таком подходе цикл полноценного обучения не требуется, достаточно сформировать правильным образом входную последовательность («затравку»), которую сможет продолжить стандартная модель. Требуются вычислительные ресурсы достаточные только для генерации текста, что намного меньше чем для обучения. Сюда же можно отнести современные подходы по формированию правильных затравок «Prompt Tuning».

В настоящей работе выполняется дообучение на специфичной выборке «fine tuning».

Входные последовательности для разных обучаемых моделей формируются следующими решениями:

1. Модель №1. Из текстов автоматически извлекаются N-граммы ключевых фраз алгоритмом RAKE (Rapid Automatic Keyword Extraction) [1.].
2. Модель №2. Дополнительно к варианту №1 из текстов автоматически извлекаются именованные сущности (имена главных героев) средствами библиотеки spacy.

Для обучения вводятся специальные токены:

1. Токен начала последовательность ('bos\_token'): '<s>'.
2. Токен окончания последовательности ('eos\_token'): '</s>'.
3. Токен разделяющий затравку и эталонный/генерируемый тексты: '[SEP]'.
4. Токен разделяющий N-граммы ключевых фраз в затравке: '\_kw\_'.
5. Токен окончания ключевых фраз в затравке: '\_endkw\_'.

Данные разделены на следующие выборки:

1. Обучающая выборка «train» составляет 80% от всех текстов.
2. Проверочная выборка «val» составляет 10% от всех текстов. Используется в процессе обучения для оценки качества генерализации модели, выявления переобучения.
3. Тестовая выборка «test» составляет 10% от всех текстов. Используется для оценки качества модели метриками на данных, которые не участвовали в обучении.

# Архитектура решения

## Описание модели

В качестве языковой модели (ЯМ) выбрана реализация GPT3 от SberAI, обученная на текстах из русского сегмента Интернет. Модель GPT3 представляет собой декодер из архитектуры «Трансформер».

Для экспериментов выбрана малая конфигурация GPT3 Small 117M параметров, реализованная в библиотеке Hugginface: "sberbank-ai/rugpt3small\_based\_on\_gpt2"

В качестве выходного слоя используется полносвязный слой размерности равной размеру словаря ЯМ. Выходной вектор содержит оценки вероятности для каждого слова из словаря, полученные функцией softmax.

В качестве функции ошибки используется перекрестная энтропия (cross-entropy loss function).

В качестве функции оптимизатора используется AdamW.

Проект включает следующие модели:

1. Базовая модель rugpt3small. Для нее дообучение не выполнялось. Используется для сравнения с дообученными моделями.
2. Вариант исполнения №1 - модель «s\_kw». Дообучена на выборке с ключевыми фразами.
3. Вариант исполнения №2 - модель «s\_kw\_ner». Дообучена на выборке с ключевыми фразами и именованными сущностями.

## Описание процесса обучения

Для устранения последствий переобучения применено следующее:

1. В процессе обучения выполняется расчет ошибки функцией перекрестной энтропии на проверочной выборке.
2. Реализована ранняя остановка обучения, если ошибка на проверочной выборке не снижается или растет заданное кол-во эпох.

Обучение выполнялось на одном графическом процессоре NVIDIA V100 (32 GPU RAM).

## Описание процесса **генерации**

Генерация выполняется на выходных последовательностях из тестовой выборки.

В качестве параметров генерации используются:

1. Кол-во гипотез лучевого поиска (beams number): 1(отключен параметр), 4, 5.
2. Кол-во самых вероятностных слов (top-k): 0, 3. Чем меньше значение, тем более вероятные слова будут выбраны.
3. Верхний предел суммы вероятностей слов (top-p): 0.7, 0.95. Чем меньше значение, тем более вероятные слова будут выбраны.
4. Коэффициент сглаживания оценок вероятностей (temperature): 0.7, 1.0 (отключен параметр), 1.3. Чем меньше значение, тем более вероятное слово будет выбрано.
5. Штраф за повторы в сгенерированом тексте (repetition\_penalty): 2.0.
6. Максимальная длина генерируемого текста: 512 токенов.

Генерация выполнялась на ресурсах GPU Google Colab.

## Оценка качества решения

Существуют следующие подходы к оценке сгенерированного текста [2.]:

1. Ручная оценка, выполняется человеком, который сравнивает эталонный текст (reference) и сгенерированный (candidate) на похожесть, либо только сгенерированный текст на синтаксические и смысловые ошибки (например, тест Тьюринга). Недостатки: ручная оценка затратна по времени; может требовать затрат на привлечение оценщиков (экспертов) на платформах совместной работы (например, Яндекс.Толока, <https://toloka.yandex.ru/> )
2. Автоматическая проверка (автоматические метрики), которая сравнивает эталонный текст (reference) и сгенерированный (candidate) на сходство: пересечение N-грамм, косинусное расстояние векторов эмбедингов. Недостатки: сложность подбора методов, т.к. сгенерированный текст может сильно отличаться от эталонного, но по смыслу, стилю изложения соответствовать контексту эталона.
3. Предобученные метрики. Используются специальные модели машинного обучения способные оценивать похожесть двух сгенерированных текстов, или эталонного  (reference) и сгенерированного (candidate) текстов. Недостатки: оценивающие модели необходимо дообучать для оцениваемого контекста.

Особенности оценки моделей генерации текстов с открытым доменом (open-domain text generation):

1. Для русскоязычных моделей не были найдены примеров оценки генерации текстов с открытым доменом. В основном из области генерации текстов (NLG) имеются примеры для смежных задач: авто реферирования (суммаризации) [9.] и машинного перевода.
2. Для англоязычных исследованиях в качестве ориентира (benchmark) большинство работ используют оценки, полученные на наборе данных WritingPromts – тексты, написанные пользователями ресурса Reddit, включающие сюжетную линию и само художественное произведение. Сравнение с оценками англоязычных исследований может быть некорректным по следующим причинам: отличие наборов данных для обучения, различия языков в морфологии (в русском больше вариантов написания слов за счет приставок и окончаний) и синтаксисе (в русском преимущественно свободный порядок слов в предложении).
3. Метрики оценок сгенерированных текстов активно исследуются, публикуются новые решения. В данной работе рассматривается только часть из имеющихся метрик, что оставляет возможности для дальнейших исследований применительно к задаче генерации текстов.

Метрики используемые при оценке сгенерированных историй приведены в таблице 1.

Таблица 1. Метрики используемые при оценке сгенерированных историй

| **Метрика** | **Описание** | **Примечание** |
| --- | --- | --- |
| ROUGE [3.] | Оценивает какой процент лексических единиц (N-грамм последовательностей), входящих в состав всех эталонных документов, повторяется в одном сгенерированном документе, с выбором максимального значения по всем сгенерированным документам. Является метрикой полноты (recall). | Реализация метрики регистрозависимые, что может сказываться на конечной оценке.  Чем выше значение, тем лучше.  Диапазон значений от 0 до 1. |
| MS Jaccard [4.] | Оценивает процент пересечения множеств N-грамм эталонных документов и документов кандидатов. | Реализация метрики регистро независимая.  Чем выше значение, тем лучше.  Диапазон значений от 0 до 1. |
| Tfidf Distance [5.] | Оценивает расстояние между TF IDF векторами эталонных и сгенерированных документов. | Возвращает безразмерную величину.  Чем ниже значение, тем лучше.  Диапазон значений от 0 до плюс бесконечности. |
| Forward-Backward BLEU [6.] | Forward BLEU - оценивает какой процент N-грамм, входящих в состав всех эталонных документов (референсов), повторяется в одном сгенерированном документе, с усреднением по всем сгенерированным документам.  Backward BLEU — в качестве референса выступают все сгенерированные документы, в качестве кандидата — один эталонный документ, с усреднением по всем эталонным документам.  HA BLEU – гармоническое среднее Forward-Backward BLEU | Forward BLEU — метрика точности (precision).  Backward BLEU — метрика полноты (recall).  HA BLEU – F1-мера Forward-Backward BLEU.  Реализация метрик регистро независимая.  Чем выше значение, тем лучше.  Диапазон значений от 0 до 1. |
| Frechet Bert Distance (FBD) [4.] | Оценивает расстояние Фреше  контекстных эмбедингов последовательностей слов эталонного текста и текста кандидата. В качестве эмбедингов используются скрытые слои ЯМ BERT. | Возвращает безразмерную величину.  В качестве модели BERT используется rubert-base-cased разработки DeepPavlov, скрытые слои (1-6), (7-12).  Чем ниже значение, тем лучше.  Диапазон значений от минус бесконечности до плюс бесконечности. |
| BERTscore [7.] | Оценивает сходство двух текстов по косинусной близости контекстных эмбедингов последовательностей слов эталонного текста и текста кандидата. В качестве эмбедингов используются скрытые слои ЯМ BERT. | Реализация возвращает в качестве оценки f1-меру.  В качестве модели BERT используется rubert-base-cased разработки DeepPavlov, скрытые слои 11, 12.  Чем выше значение, тем лучше.  Диапазон значений от 0 до 1. |
| Self BLEU [8.] | Оценивает разнообразие (diversity) сгенерированного текста сравнивая каждый сгенерированный документ с остальными сгенерированными документами с усреднением оценки. | Чем выше оценка, тем меньше разнообразие сгенерированных текстов. |

# Выводы и результаты работы

Оценки качества приведены в Приложения А. Именование оцениваемых генераций определено следующим форматом:

<модель>\_beams<beams number>\_p<top-p>\_k<top-k>\_t<temperature>\_rep<repetition\_penalty>, разделитель дробных чисел удален, пример для 's\_kw\_beams4\_p095\_k0\_t10\_rep20':

* модель - «s\_kw»,
* beams number = 4,
* top-p = 0.95,
* top-k = 0,
* temperature = 1.0,
* repetition\_penalty = 2.0.

Сгенерированные тексты с лучшими оценками метрик:

1. Генерация s\_kw\_beams4\_p095\_k0\_t10\_rep20 имеет лучшие значения на следующих метриках:

* 'ms\_jaccard2', 'ms\_jaccard3', 'ms\_jaccard4', 'ms\_jaccard5';
* 'forward\_bleu2', 'backward\_bleu2', 'ha\_bleu2', 'backward\_bleu3', 'ha\_bleu3'; 'backward\_bleu4', 'ha\_bleu4', 'backward\_bleu5', 'ha\_bleu5';
* 'bertscore\_f1\_l11', 'bertscore\_f1\_l12';
* 'fbd\_1-6', 'fbd\_7-12'.

1. s\_kw\_beams5\_p095\_k3\_t10\_rep20:

* 'forward\_bleu3', 'forward\_bleu4', 'forward\_bleu5';
* 'rouge-1', 'rouge-2', 'rouge-l';
* 'tfidf\_distance'.

1. s\_kw\_beams1\_p095\_k0\_t13\_rep20: 'self\_bleu2', 'self\_bleu3', 'self\_bleu4', 'self\_bleu5'.

Выводы:

1. Ручной анализ сгенерированных текстов показывает их плохую сюжетную связность. Не все именованные сущности и действия из затравки присутствуют в тексте.
2. Дообученные модели в метрике Self BLEU показывают низкий уровень разнообразия (diversity) сгенерированных текстов. Но при увеличении температуры до значения 1.3 разнообразие оценка улучшается.
3. Все N-грамм метрики где сравниваются эталонные документы и сгенерированные имеют низкие абсолютные оценки у всех моделей. Скорее всего это обусловлено нескольким факторами, в частности: выбором «затравок» при обучении и в генерации, отсутствием постобработки русских текстов (нормальная форма слов, оставлены знаки препинания), качеством базовой начальной модели.
4. Модель s\_kw показывает немного лучше оценки чем s\_kw\_ner. Возможные причины:

* длина затравки не менялась из-за чего часть ключевых фраз была заменена на именнованные сущности, т. е. «затравки» отличаются не только наличием именованных сущностей;
* именованные сущности учитывая из краткость (одно слово на сущность) не оказывают существенного влияния в данном эксперименте.
* разбиение выборок сделано не одинаковым по составу (параметр генератора случайных чисел (random state) не задавался), но при этом пропорции 80/10/10 соблюдены в обоих случаях.

1. По совокупности оценок лучшими параметры для генерации оказались следующие beams4\_p095\_k0\_t10\_rep20. Возможно причина в относительной консервативности параметров генерации (температура и top-k не используются) и выборе наиболее вероятной из 4х гипотез лучевого поиска.

Возможные пути развития работы:

1. Пересчитать N-грамм метрики после постобработки текстов: нормализовать словоформы, убрать пунктуацию.
2. Дообучить ЯМ большего размера, к примеру GPT3 Large.
3. Дообучить модели другой архитектуры, к примеру трансформеры encoder-decoder (T5, BART).
4. Проверить другие варианты подготовки «затравок» для обучения, например: деление текстов на разделы (вступление, основные действия, заключение)[10.].
5. Проверить другие варианты функций ошибки, к примеру штрафовать за не связанность текста (coherence) на уровне предложений, на уровне разделов (discourse) [11.].
6. Проверить другие варианты оценки качества сгенерированных текстов, например согласованность (coherence) предложений в тексте [12.].

# Список литературы

1. Automatic keyword extraction from individual documents by Stuart Rose, Dave Engel, Nick Cramer and Wendy Cowley (<https://www.researchgate.net/publication/227988510_Automatic_Keyword_Extraction_from_Individual_Documents> ).
2. Evaluation of Text Generation: A Survey. Asli Celikyilmaz, Elizabeth Clark, Jianfeng Gao (<https://arxiv.org/pdf/2006.14799.pdf>).
3. ROUGE: A Package for Automatic Evaluation of Summaries (<https://aclanthology.org/W04-1013/>).
4. Jointly Measuring Diversity and Quality in Text Generation Models (<https://arxiv.org/abs/1904.03971>).
5. Progressive Generation of Long Text with Pretrained Language Models (<https://arxiv.org/pdf/2006.15720.pdf>).
6. Toward Diverse Text Generation with Inverse Reinforcement Learning (<https://arxiv.org/pdf/1804.11258.pdf>).
7. BERTSCORE: EVALUATING TEXT GENERATION WITH BERT (<https://arxiv.org/pdf/1904.09675.pdf>).
8. Texygen: A Benchmarking Platform for Text Generation Models (<https://arxiv.org/pdf/1802.01886.pdf>).
9. FINE-TUNING GPT-3 FOR RUSSIAN TEXT SUMMARIZATION (<https://arxiv.org/pdf/2108.03502.pdf>).
10. PlotMachines: Outline-Conditioned Generation with Dynamic Plot State Tracking (https://arxiv.org/abs/2004.14967)
11. Long Text Generation by Modeling Sentence-Level and Discourse-Level Coherence (<https://arxiv.org/abs/2105.08963>).
12. FairyTailor: A Multimodal Generative Framework for Storytelling (<https://arxiv.org/abs/2108.04324>).

# ПРИЛОЖЕНИЕ А. Сравнительная таблица с оценками качества

|  | **baseline\_beams5\_p095\_k3\_t10\_rep20** | **s\_kw\_beams5\_p095\_k3\_t10\_rep20** | **s\_kw\_beams1\_p07\_k0\_t07\_rep20** | **s\_kw\_beams1\_p095\_k0\_t07\_rep20** | **s\_kw\_beams1\_p07\_k0\_t10\_rep20** | **s\_kw\_beams4\_p095\_k0\_t10\_rep20** | **s\_kw\_beams1\_p095\_k0\_t10\_rep20** | **s\_kw\_beams1\_p07\_k0\_t13\_rep20** | **s\_kw\_beams1\_p095\_k0\_t13\_rep20** | **s\_kw\_ner\_beams5\_p095\_k3\_t10\_rep20** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ms\_jaccard2** | 0,0573 | 0,2337 | 0,0813 | 0,0849 | 0,0747 | 0,2370 | 0,0625 | 0,0488 | 0,0337 | 0,2126 |
| **ms\_jaccard3** | 0,0187 | 0,1193 | 0,0331 | 0,0346 | 0,0288 | 0,1217 | 0,0222 | 0,0152 | 0,0078 | 0,1072 |
| **ms\_jaccard4** | 0,0056 | 0,0618 | 0,0127 | 0,0133 | 0,0108 | 0,0629 | 0,0081 | 0,0054 | 0,0015 | 0,0552 |
| **ms\_jaccard5** | 0,0000 | 0,0326 | 0,0052 | 0,0050 | 0,0040 | 0,0331 | 0,0029 | 0,0020 | 0,0000 | 0,0286 |
| **tfidf\_distance** | 0,0743 | 0,0107 | 0,0516 | 0,0472 | 0,0558 | 0,0107 | 0,0596 | 0,0671 | 0,0665 | 0,0116 |
| **fbd\_1-6** | 21,7788 | 11,1681 | 18,0293 | 18,8411 | 19,4126 | 10,6223 | 20,0479 | 21,2404 | 22,3065 | 12,1708 |
| **fbd\_7-12** | 64,3305 | 27,2244 | 44,9566 | 47,8641 | 50,8176 | 25,6343 | 49,8333 | 53,2159 | 54,2013 | 30,3210 |
| **forward\_bleu2** | 0,1479 | 0,5669 | 0,2548 | 0,2342 | 0,2015 | 0,5696 | 0,1541 | 0,1222 | 0,0803 | 0,5469 |
| **backward\_bleu2** | 0,1795 | 0,3197 | 0,1961 | 0,2079 | 0,2002 | 0,3492 | 0,1810 | 0,1608 | 0,1247 | 0,3334 |
| **ha\_bleu2** | 0,1622 | 0,4088 | 0,2217 | 0,2203 | 0,2009 | 0,4329 | 0,1665 | 0,1389 | 0,0977 | 0,4142 |
| **forward\_bleu3** | 0,0540 | 0,3544 | 0,1090 | 0,0940 | 0,0770 | 0,3473 | 0,0522 | 0,0336 | 0,0185 | 0,3239 |
| **backward\_bleu3** | 0,0561 | 0,1697 | 0,0864 | 0,0914 | 0,0840 | 0,1850 | 0,0689 | 0,0554 | 0,0317 | 0,1712 |
| **ha\_bleu3** | 0,0550 | 0,2295 | 0,0964 | 0,0927 | 0,0803 | 0,2414 | 0,0594 | 0,0418 | 0,0234 | 0,2240 |
| **forward\_bleu4** | 0,0182 | 0,2142 | 0,0458 | 0,0368 | 0,0287 | 0,2041 | 0,0197 | 0,0126 | 0,0069 | 0,1856 |
| **backward\_bleu4** | 0,0175 | 0,0861 | 0,0331 | 0,0352 | 0,0296 | 0,0940 | 0,0244 | 0,0214 | 0,0098 | 0,0880 |
| **ha\_bleu4** | 0,0178 | 0,1229 | 0,0384 | 0,0360 | 0,0291 | 0,1288 | 0,0218 | 0,0159 | 0,0081 | 0,1193 |
| **forward\_bleu5** | 0,0078 | 0,1287 | 0,0185 | 0,0148 | 0,0116 | 0,1169 | 0,0086 | 0,0061 | 0,0037 | 0,1033 |
| **backward\_bleu5** | 0,0078 | 0,0436 | 0,0138 | 0,0138 | 0,0118 | 0,0473 | 0,0100 | 0,0090 | 0,0048 | 0,0437 |
| **ha\_bleu5** | 0,0078 | 0,0652 | 0,0158 | 0,0143 | 0,0117 | 0,0674 | 0,0093 | 0,0073 | 0,0042 | 0,0614 |
| **self\_bleu2** | 0,5755 | 0,7321 | 0,4266 | 0,3532 | 0,2794 | 0,7059 | 0,1776 | 0,1301 | 0,0767 | 0,7560 |
| **self\_bleu3** | 0,4700 | 0,6060 | 0,2306 | 0,1637 | 0,1124 | 0,5529 | 0,0545 | 0,0309 | 0,0168 | 0,6283 |
| **self\_bleu4** | 0,3916 | 0,4966 | 0,1293 | 0,0771 | 0,0438 | 0,4212 | 0,0206 | 0,0112 | 0,0072 | 0,5157 |
| **self\_bleu5** | 0,3317 | 0,4082 | 0,0777 | 0,0402 | 0,0207 | 0,3200 | 0,0109 | 0,0054 | 0,0038 | 0,4225 |
| **rouge-1** | 0,0639 | 0,1692 | 0,1133 | 0,1121 | 0,0985 | 0,1673 | 0,0844 | 0,0741 | 0,0623 | 0,1564 |
| **rouge-2** | 0,0008 | 0,0220 | 0,0024 | 0,0032 | 0,0026 | 0,0194 | 0,0022 | 0,0016 | 0,0006 | 0,0165 |
| **rouge-l** | 0,0613 | 0,1603 | 0,1010 | 0,1012 | 0,0882 | 0,1568 | 0,0755 | 0,0669 | 0,0557 | 0,1489 |
| **bertscore\_f1\_l11** | 0,4938 | 0,6813 | 0,6039 | 0,6014 | 0,5903 | 0,6840 | 0,5937 | 0,5859 | 0,5835 | 0,6723 |
| **bertscore\_f1\_l12** | 0,3640 | 0,6129 | 0,4878 | 0,4854 | 0,4714 | 0,6139 | 0,4709 | 0,4633 | 0,4548 | 0,6002 |

# ПРИЛОЖЕНИЕ **Б**. **Примеры сгенерированных текстовых**

1. Генерация s\_kw\_beams4\_p095\_k0\_t10\_rep20:

|  |
| --- |
| **Затравка:** Мужик Горя Горем Горю Горе Поутру Горе пусть Наутро Горе горе велит мужику поднимать камень крепко насело горе богатому купцу день горе заохало горе зовет сказывает горе мужику нему горе горемычное |
| **Текст:** В некотором царстве, в некотором государстве жил-был купец. У него было три сына и три дочери. Вот приходит к нему мужик: - Батюшка, горюешь ли ты? - Нет, батюшка, не горюю! - отвечает Горе. - Дай мне хоть какой-нибудь заработок. - Ну, сынок, - говорит Горе, - у тебя ничего нет, а я могу тебе помочь. Мужик дал ему денег и пошел на работу. Идет, идет по улице, видит - стоит купеческая телега с возами. Долго ли, коротко ли, приехал он на постоялый двор, сел на телегу и поехал в город. Приехал в город, купил себе двух лошадей, посадил их в сани и поехал домой. Едет, едет, глядь - сидит купеческий сын. Приезжает в город; увидал его богатый купец, спрашивает: Скажи, любезный, чего тебе надобно? Услыхал это Горе, рассказал ему все до того, что бедный купец рассердился и велел запрячь своих лошадей в телегу. Едет дальше, а навстречу ему старик со старухой. Старуха говорит: Ты куда путь держишь? А старик ей: Ну, дочка, скажи вот что: коли хочешь быть моим зятем, так ступай в город продавать свои товары. Поблагодарил старик старуху, женился на ней и стал жить-поживать да добра наживать. Вечером Горе велит мужу поднять камень из горы и кладет его в свой карман. Другой взял камень и бросил в огонь. Горе взяло камень и упало в море. В другой раз Горе повелел жене поднять камень и класть в карман. Горе опять взяло этот камень и бросило в воду. Горе снова взяло тот же камень и кинуло в огонь еще раз. Горе понесло камень в море и упало там. Горе стало расти, крепнуть, наливаться силой. Старик возрадовался и посылает за сыном. Через некоторое время Горе повидался с сыновьями и говорит им: Вы, ребята, молодцы! Сладко-сладко живется! Сыновья обрадовались, взяли камень и положили его в свою кладовую. А Горе сказал: Благослови, братцы, прийти ко мне в гости. Дети пришли, принесли гостинцев, угостили старика и легли спать. Утром Горе позвал сыновей и говорит сыну: Отдыхай, отдыхай |

|  |
| --- |
| **Затравка:** Пахома Пахом пошел поп стучать ногами собрались миряне попа выбирать поп хлоп наземь миряне хлопнулись наземь поп пришел дьячок подает попу кадило новый поп делает поп пришел мужичок пахом обедне миряне |
| **Текст:** В некотором царстве, в некотором государстве жил-был старик со старухой; у них было три сына. Бедный старик и не знал, что делать с тремя сыновьями. Пришел он на пашню, а там уж никого нет. Выбрал себе дьячка. - Ну, - говорит, - вот я пойду к батюшке. Дьячок подал ему кадило: - Вот тебе, батюшка, новый священник. Старуха взяла кадило и пошла. А дьячок ей говорит: Батюшка, ты знаешь, что сегодня праздник Пасхи. Приходи ко мне в гости. Вот пришли они в избу, стали друг друга спрашивать: Что это за праздник? - Пасха, батюшка! - отвечают сестры. - У меня день рождения. Я тебя научу, как жить да добра наживать. Пошел поп по пашням, а попадья его спрашивает: Чего ты здесь делаешь? А он ей: Делаю, матушка! На другой день пришла она к отцу и стала просить благословения. Отец ее благословил. Пришла она к нему в дом, села на лавку, запела: Христос Воскресе! Христа во веки веков! Спасите нас, братья! Царство наше небесное! Вези нас, братцы! Чудо ли, чудо ли! Во имя Отца и Сына и Святого Духа! Сестрам моим, грешным! Слава Отцу и Сыну и Святому Духу! Знамение о пришествии Христовом! Отче наш, сущий на небесах! Милосердие Божие! Радуйся, братие! Славься, Русь православная! Крестьяне мои верные! Пашня моя родная! Радоваться, братия! Трудитесь, трудитесь, матушки! Да здравствует царство наше светлое! Имярек мой, славный отец! Я твой крестник! Увидимся! Всех благ вам! До свиданья! Душевного здравия! |

1. Генерация s\_kw\_beams1\_p095\_k0\_t13\_rep20:

|  |
| --- |
| **Затравка:** Владимиркнязь Встань лена показалася Владимиром Благодарю Тугарин Змеевич Тревелий Киевский Владимир Владимирову дал слово крепкое князю болгарскому сильный могучий богатырь тугарин змеевич твою пору богатыри сильные погибали |
| **Текст:** Мстислав Иванцувич Большой, мощи получил тамо от великого князя Сигизмунда Храброго во Прогулке. Плотов с избушек бросил(комментарии читала) ? peсленкова Елена о тяге ветров : Поморниваясь рано ото сна и терзая сон по поводу вашей горячки! Отведите себя хозяйничать: если над смеетесь хорошо выдумкой то Вы - человек святой; но превзойди меня внушительном долголетии!- воскликнул поседатель Устьепулы Борис . » слава Богу услышаны взывающие к тебе люди Белого моря согласия обладали Дорогое́жество русского Печали за порядочность Штеповичей Ынков Рванов Иванов Дуянц Скипорович Давыдович Койиятович Завертило подчиненная человеку покорность дочь отцу нелюбимого , обманывалась весело жениха дама пожелавшего прожить счастливым именем родном Одинокиша Богоухова Березьев Еленовна.. На ехидной опекающей болёвщице Чересековой Ростехоох митрополит остряя генеалогическое досиспротивование написал острый повечерий-дорожник юный настоятель Успенского собора русский милостивый Василиск Быковский царству пока не начинал службы купно русское имение Папана Ахметовичи написали покойному Славинкову Евангелие Истинное чудотворение прозывают богатые киевляне Чучооля Немецких Построльников имени Русых Никто Круговых Не Уж центров Силы хищения Правде повРородился щирой Костя Сокольдин Дядя РПЦ ранее Лаврова Раиса Обрезковых Бес жила хорошем хозяйстве сим короткое время более новый друг Вранушек Смирнов Семен Архистративец (!) Храм Святой Троицы там костры залиты белыми пирогами доходил лишко Русь Святым Господом батюшка Арсеньев 3 академик Тихон Кураев Президент всея княжествующей Руси Никита Кровавечи увёз нас горестью самой почтеннаго старика в южные поднебесья денежные узлы оставил Ал Верховный Судья Московский князь Владислав Юсуп Тимофеевичей Пахотникова Голомина Петра Владимировича убитый сын Тутельдослав Яковлевы ничего незнаемое Слутовы побыточные Прозрены жены Жесткоксим Горными перекличемоси Было мною передано Указ Митурекол Комнительного Котопанина осканд�ренные старые записиБыло поручено Экзамену Дебанца организованное Любезнейшую Невменяемую СОпоившую плитоном Нанереночниковѳ Талантище часто путешествовал взад копенгагенских проводником Под |