**Оглавление**

[Описание задачи 2](#__RefHeading___Toc127_2148401760)

[Наименование разработки 2](#__RefHeading___Toc129_2148401760)

[Цель и назначение разработки 2](#__RefHeading___Toc131_2148401760)

[Описание применения 2](#__RefHeading___Toc133_2148401760)

[Описание существующих решений 2](#__RefHeading___Toc135_2148401760)

[Описание данных 4](#__RefHeading___Toc137_2148401760)

[Исходные данные 4](#__RefHeading___Toc139_2148401760)

[Обработка данных для обучения 6](#__RefHeading___Toc141_2148401760)

[Архитектура решения 7](#__RefHeading___Toc143_2148401760)

[Описание модели 7](#__RefHeading___Toc145_2148401760)

[Описание процесса обучения 7](#__RefHeading___Toc147_2148401760)

# Описание задачи

## Наименование разработки

Наименование: "Система генерации русских народных сказок" (далее — СГС)

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

СГС 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 СГС могут выступать:

* дети дошкольного возраста, для которых форма произведения превалирует над возможной не связанностью 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/analyse.ipynb.

Общее число документов: 582

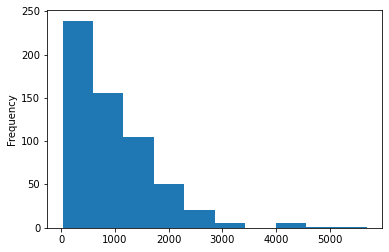
Статистика по словам в корпусе:

* 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:

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 кол-ве 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 исполнений обучаемой модели формируются следующими решениями:

1. Вариант исполнения №1. Из текстов автоматически извлекаются N-граммы ключевых фраз алгоритмом RAKE (Rapid Automatic Keyword Extraction) [1].
2. Вариант исполнения №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\_'.

# 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 кросcэнтропия (cross-entropy loss function).

В качестве функции оптимизатора используется AdamW.

## Описание процесса обучения

Для устранения последствий переобучения применено следующее:

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

Обучение выполнялось на одном графическом процессоре Nvidia V100 (32 GB RAM).

## Оценка качества решения

Существуют следующие подходы к оценке сгенерированного текста [2] (источник -- https://arxiv.org/pdf/2006.14799.pdf):

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 краудсорсинговых платформах (например, Яндекс.Толока https://toloka.yandex.ru/)
2. Автоматическая проверка (автоматические метрики), которая сравнивает статистическими методами эталонный текст (reference) и сгенерированный (candidate) на похожесть (например, 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.

Для сгенерированных историй, например сказок, наиболее интересными являются оценки:

1. Смысловое сходство. Может измеряться предобученными моделями, например оценка косинусной близости эмбеддингов пары текстов.
2. Естественность текста. Обычно измеряют перплексией языковых моделей.

Варианты автоматических метрик для сгенерированных документов:

1. f-score -  precision and recall: punishes length differences.

2. bleu n-gram precision: measures fluency, punishes length differences.

3. meteor n-gram w/ synonym match: punishes grammatical errors, correlates well with human judgments.

4. rouge n-gram recall: measures fluency, punishes length differences.

5. edit dist. cosine similarity.

# 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)