Физтех-Школа Прикладной математики и информатики (ФПМИ) МФТИ

Text Summarization

Всем привет! Сегодня мы познакомимся с задачей суммаризации текста на примере генерации "сжатых" новостей. Рассмотрим некоторые базовые решения и познакомимся с архитектурами нейросетей для решения задачи. Датасет: gazeta.ru

Ноутбук создан на основе семинара Гусева Ильи на кафедре компьютерной лингвистики ABBYY МФТИ.

Загрузим датасет и необходимые библиотеки

```
!wget -q https://www.dropbox.com/s/43l702z5a5i2w8j/gazeta_train.txt
!wget -q https://www.dropbox.com/s/k2egt3sug0hb185/gazeta_val.txt
!wget -q https://www.dropbox.com/s/3gki5n5djs9w0v6/gazeta_test.txt
```

```
!pip -q install razdel networkx pymorphy2 nltk rouge==0.3.1 summa
```

```
     |████████████████████████████████| 61kB 3.1MB/s
     |████████████████████████████████| 61kB 5.7MB/s
     |████████████████████████████████| 8.2MB 7.6MB/s
  Building wheel for summa (setup.py) ... done
```

Dataset

Посмотрим на то, как устроен датасет

```
!head -n 1 gazeta_train.txt
!cat gazeta_train.txt | wc -l
!cat gazeta_val.txt | wc -l
!cat gazeta_test.txt | wc -l
```

```
{"url": "https://www.gazeta.ru/financial/2011/11/30/3852658.shtml", "text": "«По итогам 2011 года чистый отток может составить примерно $80 млрд, в следующем году — около $20 млрд. При этом мы ожидаем, что со второго полугодия 2012
52400
5265
5770
```

```
import json
import random

def read_gazeta_records(file_name, shuffle=True, sort_by_date=False):
    assert shuffle != sort_by_date
    records = []
    with open(file_name, "r") as r:
        for line in r:
            records.append(json.loads(line))
    if sort_by_date:
        records.sort(key=lambda x: x["date"])
    if shuffle:
        random.shuffle(records)
    return records
```

```
train_records = read_gazeta_records("gazeta_train.txt")
val_records = read_gazeta_records("gazeta_val.txt")
test_records = read_gazeta_records("gazeta_test.txt")
```

```
from nltk.translate.bleu_score import corpus_bleu
from rouge import Rouge

def calc_scores(references, predictions, metric="all"):
    print("Count:", len(predictions))
    print("Ref:", references[-1])
    print("Hyp:", predictions[-1])

    if metric in ("bleu", "all"):
        print("BLEU: ", corpus_bleu([[r] for r in references], predictions))
    if metric in ("rouge", "all"):
        rouge = Rouge()
        scores = rouge.get_scores(predictions, references, avg=True)
        print("ROUGE: ", scores)
```

Extractive RNN

BPE

Для начала сделаем BPE токенизацию

```
!pip install youtokentome
```

```
Collecting youtokentome
  Downloading https://files.pythonhosted.org/packages/c8/1c/224cdc3d9a32ed706c8fb1f30b491be6ea5da114ff4edc174014cc24fa43/youtokentome-1.0.6-cp37-cp37m-manylinux2010_x86_64.whl (1.7MB)
     |████████████████████████████████| 1.7MB 4.2MB/s
Requirement already satisfied: Click>=7.0 in /usr/local/lib/python3.7/dist-packages (from youtokentome) (7.1.2)
Installing collected packages: youtokentome
Successfully installed youtokentome-1.0.6
```

```
import youtokentome as yttm

def train_bpe(records, model_path, model_type="bpe", vocab_size=10000, lower=True):
    temp_file_name = "temp.txt"
    with open(temp_file_name, "w") as temp:
```

```
    with open(temp_file_name, 'w') as temp:
        for record in records:
            text, summary = record['text'], record['summary']
            if lower:
                summary = summary.lower()
                text = text.lower()
            if not text or not summary:
                continue
            temp.write(text + "\n")
            temp.write(summary + "\n")
    yttm.BPE.train(data=temp_file_name, vocab_size=vocab_size, model=model_path)

train_bpe(train_records, "BPE_model.bin")
```

Словарь

Составим словарь для индексации токенов

```
bpe_processor = yttm.BPE('BPE_model.bin')
vocabulary = bpe_processor.vocab()
```

Кэш oracle summary

Закэшируем oracle summary, чтобы не пересчитывать их каждый раз

```
from rouge import Rouge
import razdel
from tqdm.notebook import tqdm

import copy

def build_oracle_summary_greedy(text, gold_summary, calc_score, lower=True, max_sentences=30):
    '''
    Жадное построение oracle summary
    '''
    gold_summary = gold_summary.lower() if lower else gold_summary
    # Делим текст на предложения
    sentences = [sentence.text.lower() if lower else sentence.text for sentence in razdel.sentenize(text)][:max_sentences]
    n_sentences = len(sentences)
    oracle_summary_sentences = set()

    score = -1.0
    summaries = []
    for _ in range(n_sentences):
        for i in range(n_sentences):
            if i in oracle_summary_sentences:
                continue
            current_summary_sentences = copy.copy(oracle_summary_sentences)
            # Добавляем какое-то предложения к уже существующему summary
            current_summary_sentences.add(i)
            current_summary = " ".join([sentences[index] for index in sorted(list(current_summary_sentences))])
            # Считаем метрики
            current_score = calc_score(current_summary, gold_summary)
            summaries.append((current_score, current_summary_sentences))
        # Если получилось улучшить метрики с добавлением какого-либо предложения, то пробуем добавить ещё
        # Иначе на этом заканчиваем
        best_summary_score, best_summary_sentences = max(summaries)
        if best_summary_score <= score:
            break
        oracle_summary_sentences = best_summary_sentences
        score = best_summary_score
    oracle_summary = " ".join([sentences[index] for index in sorted(list(oracle_summary_sentences))])
    return oracle_summary, oracle_summary_sentences

def calc_single_score(pred_summary, gold_summary, rouge):
    return rouge.get_scores([pred_summary], [gold_summary], avg=True)['rouge-2']['f']

def add_oracle_summary_to_records(records, max_sentences=30, lower=True, nrows=1000):
    rouge = Rouge()
    for i, record in tqdm(enumerate(records)):
        if i >= nrows:
            break
        text = record["text"]
        summary = record["summary"]

        summary = summary.lower() if lower else summary
        sentences = [sentence.text.lower() if lower else sentence.text for sentence in razdel.sentenize(text)][:max_sentences]
        oracle_summary, sentences_indicies = build_oracle_summary_greedy(text, summary, calc_score=lambda x, y: calc_single_score(x, y,
                                                                          lower=lower, max_sentences=max_sentences)
        record["sentences"] = sentences
        record["oracle_sentences"] = list(sentences_indicies)
        record["oracle_summary"] = oracle_summary

    return records[:nrows]

ext_train_records = add_oracle_summary_to_records(train_records, nrows=2048)
ext_val_records = add_oracle_summary_to_records(val_records, nrows=256)
ext_test_records = add_oracle_summary_to_records(test_records, nrows=256)
```

```
2048/? [05:58<00:00, 2.75it/s]

255/? [00:53<00:00, 3.70it/s]

256/? [00:57<00:00, 3.14it/s]
```

Составление батчей

```
import torch
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
```

```
import random
import math
import razdel
import torch
import numpy as np
from rouge import Rouge


class BatchIterator():
    def __init__(self, records, vocabulary, batch_size, bpe_processor, shuffle=True, lower=True, max_sentences=30, max_sentence_length=
        self.records = records
        self.num_samples = len(records)
        self.batch_size = batch_size
        self.bpe_processor = bpe_processor
        self.shuffle = shuffle
        self.batches_count = int(math.ceil(self.num_samples / batch_size))
        self.lower = lower
        self.rouge = Rouge()
        self.vocabulary = vocabulary
        self.max_sentences = max_sentences
        self.max_sentence_length = max_sentence_length
        self.device = device

```

```
    def __len__(self):
        return self.batches_count

    def __iter__(self):
        indices = np.arange(self.num_samples)
        if self.shuffle:
            np.random.shuffle(indices)

        for start in range(0, self.num_samples, self.batch_size):
            end = min(start + self.batch_size, self.num_samples)
            batch_indices = indices[start:end]

            batch_inputs = []
            batch_outputs = []
            max_sentence_length = 0
            max_sentences = 0
            batch_records = []

            for data_ind in batch_indices:

                record = self.records[data_ind]
                batch_records.append(record)
                text = record["text"]
                summary = record["summary"]
                summary = summary.lower() if self.lower else summary

                if "sentences" not in record:
                    sentences = [sentence.text.lower() if self.lower else sentence.text for sentence in razdel.sentenize(text)][:self.m
                else:
                    sentences = record["sentences"]
                max_sentences = max(len(sentences), max_sentences)

                # номера предложений, которые в нашем саммари
                if "oracle_sentences" not in record:
                    calc_score = lambda x, y: calc_single_score(x, y, self.rouge)
                    sentences_indicies = build_oracle_summary_greedy(text, summary, calc_score=calc_score, lower=self.lower, max_sentenc
                else:
                    sentences_indicies = record["oracle_sentences"]

                # inputs - индексы слов в предложении
                inputs = [bpe_processor.encode(sentence)[:self.max_sentence_length] for sentence in sentences]
                max_sentence_length = max(max_sentence_length, max([len(tokens) for tokens in inputs]))

                # получение метки класса предложения
                outputs = [int(i in sentences_indicies) for i in range(len(sentences))]
                batch_inputs.append(inputs)
                batch_outputs.append(outputs)

            tensor_inputs = torch.zeros((self.batch_size, max_sentences, max_sentence_length), dtype=torch.long, device=self.device)
            # we add index 2 for padding
            # YOUR CODE
            tensor_outputs = torch.full((self.batch_size, max_sentences), fill_value=2.0, dtype=torch.float32, device=self.device)

            for i, inputs in enumerate(batch_inputs):
                for j, sentence_tokens in enumerate(inputs):
                    tensor_inputs[i][j][:len(sentence_tokens)] = torch.tensor(sentence_tokens, dtype=torch.int64)

            for i, outputs in enumerate(batch_outputs):
                tensor_outputs[i][:len(outputs)] = torch.LongTensor(outputs)

            tensor_outputs = tensor_outputs.long()

            yield {
                'inputs': tensor_inputs,
                'outputs': tensor_outputs,
                'records': batch_records
            }
```

```
train_iterator = BatchIterator(ext_train_records, vocabulary, 32, bpe_processor, device=device)
val_iterator = BatchIterator(ext_val_records, vocabulary, 32, bpe_processor, device=device)
test_iterator = BatchIterator(ext_test_records, vocabulary, 32, bpe_processor, device=device)
```

Extractor - SummaRuNNer

https://arxiv.org/pdf/1611.04230.pdf

Homework

- В данной реализации в `outputs` в качестве padding используется индекс 0. Измените в функции __iter__ индекс padding, чтобы он не совпадал с классом 0 или 1, например, 2.
- В качестве criterion используйте `CrossEntropyLoss` вместо `BCEWithLogitsLoss`
- Из-за смены criterion, вы уже должны подавать на вход criterion ни одно число, а logits для каждого класса. Перед подачей logits вы можете отфильтровать предсказания для класса padding. В этом пункте вам придется изменять файл `train_model.py`, а именно функциии `train` и `evaluate`.
- Используйте два варианта обучения: с весами в `CrossEntropyLoss` и без
- Также сравните `inference`, когда вы ранжируете logits, и когды вы выбирате предложения, у котрых logits > 0, в двух вариантах обучения.
- Реализуйте дополнительно характеристику предложения `novelty`. Как влияет добавление `novelty` на качество summary?
- Постарайтесь улучшить качество модели, полученной на семинаре: $BLEU \approx 0.45$

```
import numpy as np

import torch
import torch.nn as nn
import torch.nn.functional as F
import torch.optim as optim
from torch.autograd import Variable

from torch.nn.utils.rnn import pack_padded_sequence as pack
from torch.nn.utils.rnn import pad_packed_sequence as unpack

class SentenceEncoderRNN(nn.Module):
    def __init__(self, input_size, embedding_dim, hidden_size, n_layers=3, dropout=0.3, bidirectional=True):
        super().__init__()

        num_directions = 2 if bidirectional else 1
        assert hidden_size % num_directions == 0
        hidden_size = hidden_size // num_directions

        self.embedding_dim = embedding_dim
        self.input_size = input_size
        self.hidden_size = hidden_size
        self.n_layers = n_layers
        self.dropout = dropout
        self.bidirectional = bidirectional

        self.embedding_layer = nn.Embedding(input_size, embedding_dim)
        self.rnn_layer = nn.LSTM(embedding_dim, hidden_size, n_layers, dropout=dropout, bidirectional=bidirectional, batch_first=True)
        self.dropout_layer = nn.Dropout(dropout)

    def forward(self, inputs, hidden=None):
```

```
        embedded = self.dropout_layer(self.embedding_layer(inputs))
        outputs, _ = self.rnn_layer(embedded, hidden)
        sentences_embeddings = torch.mean(outputs, 1)
        # [batch_size, hidden_size]
        return sentences_embeddings

class SentenceTaggerRNN(nn.Module):
    def __init__(self,
                 vocabulary_size,
                 use_content=True,
                 use_salience=True,
                 use_novelty=True,
                 token_embedding_dim=128,
                 sentence_encoder_hidden_size=256,
                 hidden_size=256,
                 bidirectional=True,
                 sentence_encoder_n_layers=2,
                 sentence_encoder_dropout=0.3,
                 sentence_encoder_bidirectional=True,
                 n_layers=2,
                 dropout=0.3):

        super().__init__()

        num_directions = 2 if bidirectional else 1
        assert hidden_size % num_directions == 0
        hidden_size = hidden_size // num_directions

        self.hidden_size = hidden_size
        self.n_layers = n_layers
        self.dropout = dropout
        self.bidirectional = bidirectional

        self.sentence_encoder = SentenceEncoderRNN(vocabulary_size, token_embedding_dim,
                                                   sentence_encoder_hidden_size, sentence_encoder_n_layers,
                                                   sentence_encoder_dropout, sentence_encoder_bidirectional)

        self.rnn_layer = nn.LSTM(sentence_encoder_hidden_size, hidden_size, n_layers, dropout=dropout,
                                 bidirectional=bidirectional, batch_first=True)

        self.dropout_layer = nn.Dropout(dropout)
        self.content_linear_layer = nn.Linear(hidden_size * 2, 1)
        self.document_linear_layer = nn.Linear(hidden_size * 2, hidden_size * 2)
        self.salience_linear_layer = nn.Linear(hidden_size * 2, hidden_size * 2)
        self.novelty_linear_layer = nn.Linear(hidden_size * 2, hidden_size * 2)
        self.tanh_layer = nn.Tanh()

        self.use_content = use_content
        self.use_salience = use_salience
        self.use_novelty = use_novelty

    def forward(self, inputs, hidden=None):
        # parameters of the probability
        content = 0
        salience = 0
        novelty = 0

        # [batch_size, seq num, seq_len]
        batch_size = inputs.size(0)
        sentences_count = inputs.size(1)
        tokens_count = inputs.size(2)
        inputs = inputs.reshape(-1, tokens_count)
        # [batch_size * seq num, seq_len]

        embedded_sentences = self.sentence_encoder(inputs)
        embedded_sentences = self.dropout_layer(embedded_sentences.reshape(batch_size, sentences_count, -1))
        # [batch_size *  seq num, seq_len, hidden_size] -> [batch_size, seq num, hidden_size]

        outputs, _ = self.rnn_layer(embedded_sentences, hidden)
        # [batch_size, seq num, hidden_size]

        document_embedding = self.tanh_layer(self.document_linear_layer(torch.mean(outputs, 1)))
        # [batch_size, hidden_size]

        # W * h^T
        if self.use_content:
            content = self.content_linear_layer(outputs).squeeze(2) # 1-representation
        # [batch_size, seq num]

        # h^T * W * d
        if self.use_salience:
            salience = torch.bmm(outputs, self.salience_linear_layer(document_embedding).unsqueeze(2)).squeeze(2) # 2-representation
        # [batch_size, seq num, hidden_size] * [batch_size, hidden_size, 1] = [batch_size, seq num, ]

        if self.use_novelty:
            # at every step add novelty to prediction of the sentence
            predictions = content + salience

            # 0) initialize summary_representation and novelty by zeros
            # YOUR CODE
            summary_representation = torch.zeros(batch_size, self.hidden_size * 2).to(device) #[batch_size, hidden_size]
            novelty = torch.zeros(batch_size, sentences_count)

            for sentence_num in range(sentences_count):

                # 1) take sentence_num_state from outputs(representation of the sentence with number sentence_num)
                # 2) calculate novelty for current sentence
                # 3) add novelty to predictions
                # 4) calculcate probability for current sentence
                # 5) add sentence_num_state with the weight which is equal to probability to summary_representation

                # YOUR CODE
                sentence_num_state = outputs[:, sentence_num, :] #1

                novelty = torch.bmm(outputs, self.novelty_linear_layer(torch.tanh(summary_representation)).unsqueeze(2)).squeeze(2) #2
                # [batch_size, seq num, hidden_size] * [batch_size, hidden_size, 1] = [batch_size, seq num, ]

                predictions[:, sentence_num] += novelty[:, sentence_num] #3

                probability = torch.sigmoid(predictions)[:, sentence_num].unsqueeze(1) #4

                summary_representation += probability * sentence_num_state  #5

        return content + salience + novelty
```

Model

$$P\left(y_j = 1 \mid \mathbf{h}_j, \mathbf{s}_j, \mathbf{d}\right) = \sigma\left(W_c \mathbf{h}_j + \mathbf{h}_j^T W_s \mathbf{d}\right)$$

```
#!gdown https://drive.google.com/uc?id=1MiS_iczALcyF7zGDPY6niyeD82P0_PBH -O train_model.py

from google.colab import drive
drive.mount('/content/drive')

!cp /content/drive/MyDrive/hw_text_summarization/train_model.py .

```

```
import train_model
import imp
imp.reload(train_model)
from train_model import train_with_logs
```

```
Mounted at /content/drive
```

```
all_counts = torch.zeros(3)
for batch in iter(train_iterator):
    all_counts += batch['outputs'].view(-1).unique(return_counts=True)[1].cpu()

class_0_count = all_counts[0]
class_1_count = all_counts[1]

print("class_0_count=", class_0_count)
print("class_1_count=", class_1_count)

weights = torch.tensor([class_1_count / (class_0_count + class_1_count), class_0_count / (class_0_count + class_1_count)])
weights = weights.to(device)
weights
```

```
class_0_count= tensor(53811.)
class_1_count= tensor(4620.)
tensor([0.0791, 0.9209], device='cuda:0')
```

```
N_EPOCHS = 20
CLIP = 1

def train_all(use_class_weights, N_EPOCHS, CLIP, lr=1e-3):
    optimizer = optim.Adam(model.parameters(), lr)
    if use_class_weights:
        # weights depend on the number of objects of class 0 and 1
        criterion = nn.CrossEntropyLoss(weight=weights, ignore_index=2)
    else:
        criterion = nn.CrossEntropyLoss(ignore_index=2)
    train_with_logs(model, train_iterator, val_iterator, optimizer, criterion, N_EPOCHS, CLIP)
```

Inference

```
from train_model import punct_detokenize, postprocess

def top_k_inference_summarunner(model, iterator, top_k=3):
    print("top_k=", top_k)

    references = []
    predictions = []

    model.eval()
    for batch in test_iterator:

        preds = model(batch['inputs'])
        sum_in = torch.argsort(preds, dim=1)[:, -top_k:]

        for i in range(len(batch['outputs'])):

            summary = batch['records'][i]['summary'].lower()
            pred_summary = ' '.join([batch['records'][i]['sentences'][ind] for ind in sum_in.sort(dim=1)[0][i]])

            summary, pred_summary = postprocess(summary, pred_summary)

            references.append(summary)
            predictions.append(pred_summary)

    calc_scores(references, predictions)
```

```
def threshold_inference_summarunner(model, iterator, threshold=0):
    print("threshold", threshold)

    references = []
    predictions = []

    model.eval()
    for batch in test_iterator:

        preds = model(batch['inputs'])

        for i in range(len(batch['outputs'])):
            sentences = (preds[i] > threshold).nonzero(as_tuple=True)[0].unique()

            summary = batch['records'][i]['summary'].lower()
            pred_summary = ' '.join([batch['records'][i]['sentences'][ind] for ind in sentences if ind < len(batch['records'][i]['sent

            if len(pred_summary) == 0:
                pred_summary = ' '.join(batch['records'][i]['sentences'])

            summary, pred_summary = postprocess(summary, pred_summary)

            references.append(summary)
            predictions.append(pred_summary)

    calc_scores(references, predictions)
```

```
vocab_size = len(vocabulary)
```

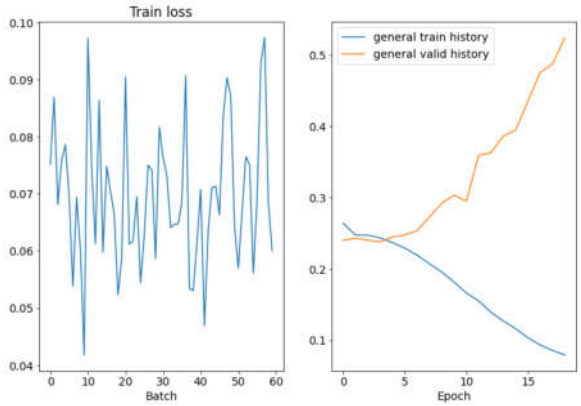
Обучим модель без весов и оценим ее качество

```
model = SentenceTaggerRNN(vocab_size, use_novelty=False).to(device)

train_all(False, N_EPOCHS, CLIP)

model.load_state_dict(torch.load('best-val-model.pt'))

top_k_inference_summarunner(model, test_iterator, 1)
top_k_inference_summarunner(model, test_iterator, 2)
top_k_inference_summarunner(model, test_iterator, 3)
top_k_inference_summarunner(model, test_iterator, 5)
threshold_inference_summarunner(model, test_iterator, 0.2)
threshold_inference_summarunner(model, test_iterator, 0.3)
threshold_inference_summarunner(model, test_iterator, 0.35)
threshold_inference_summarunner(model, test_iterator, 0.4)
threshold_inference_summarunner(model, test_iterator, 0.5)
```




```
Epoch: 20 | Time: 0m 8s
        Train Loss: 0.071 | Train PPL:   1.073
         Val. Loss: 0.515 |  Val. PPL:   1.673
top_k= 1
Count: 256
Ref: убийство пентагоном главы спецназа « аль-кудс » ирана касема сулеймани может серьезно обострить ситуацию в ближневосточном регионе — к такому выводу пришли президенты россии владимир путин и франции эммануэль макрон в телефонно
Hyp: президенты россии владимир путин и франции эммануэль макрон в телефонном разговоре , прошедшем в пятницу , 3 января , выразили озабоченность по поводу ликвидации американскими военными командующего силами специального назначени
BLEU:  0.19516170557764109
ROUGE:  {'rouge-1': {'f': 0.27999793835109166, 'p': 0.44137925025270525, 'r': 0.21734463160292056}, 'rouge-2': {'f': 0.13960692179339343, 'p': 0.2252523034115662, 'r': 0.10789084125917864}, 'rouge-l': {'f': 0.20939310777214343, 'p':
top_k= 2
Count: 256
Ref: сахалинская и магаданская области были названы самыми « пьющими » регионами россии за 2019 год . в топ-5 также вошли республика коми , чукотка и карелия . эксперты объясняют уровень потребления водки не только холодным климатом
Hyp: названы регионы , где пьют больше всего водки . среди лидеров рейтинга — сахалинская и магаданская области ( 12 и 11,5 л водки на душу населения в год ) .
BLEU:  0.42867053203944355
ROUGE:  {'rouge-1': {'f': 0.30897024493067454, 'p': 0.34477091540718274, 'r': 0.29687727248350937}, 'rouge-2': {'f': 0.14464002738736406, 'p': 0.15990398437415757, 'r': 0.1426943139813192}, 'rouge-l': {'f': 0.26112008219834293, 'p':
top_k= 3
Count: 256
Ref: актеры колин фаррелл и энди серкис могут сыграть в новом фильме о бэтмене . по данным сми , первый рассматривается на роль пингвина , тогда как второй — дворецкого альфреда . кроме того , к звездному составу может присоединитьс
Hyp: актеры колин фаррелл и энди серкис ведут переговоры , чтобы сняться в новом фильме о супергерое бэтмене , передает variety со ссылки на источники . по информации издания , фаррелл может сыграть пингвина . до него в роли злодея
BLEU:  0.43134285190491656
ROUGE:  {'rouge-1': {'f': 0.30227019251814835, 'p': 0.28166637271609773, 'r': 0.34496439737600176}, 'rouge-2': {'f': 0.13488600921636967, 'p': 0.1224499666258892, 'r': 0.16186738536338266}, 'rouge-l': {'f': 0.2566365952951387, 'p':
top_k= 5
Count: 256
Ref: часы судного дня , символизирующие самые опасные периоды времени для человечества , перевели на 20 секунд ближе к полуночи . причины этому : возросшая опасность ядерной войны , катастрофическое изменения климата из-за неразумно
Hyp: условные часы судного дня , символизирующие самые опасные периоды времени для человечества , перевели на 20 секунд ближе к полуночи — условной точке конца мира . причины такого решения поясняются в заявлении , опубликованном в
BLEU:  0.3066300864054362б
ROUGE:  {'rouge-1': {'f': 0.2769631515044622, 'p': 0.2120418766243586, 'r': 0.4198078951551332}, 'rouge-2': {'f': 0.11668355888946855, 'p': 0.08603368225637263, 'r': 0.1936336930041064}, 'rouge-l': {'f': 0.20932158642920198, 'p': 0.
threshold 0.2
Count: 256
Ref: поддержка малого и среднего предпринимательства в россии в этом году получила новый толчок в рамках национального проекта . до 2024 года планируется увеличить долю мсп в ввп страны с нынешних 22,3 % до 32,5 % , а также число за
Hyp: в россии планируется привлечь как можно больше граждан в предпринимательскую деятельность . такова одна из целей нацпроекта « малое и среднее предпринимательство » . на популяризацию предпринимательства среди населения до 2024
BLEU:  0.08865525571511547
ROUGE:  {'rouge-1': {'f': 0.19200576339545652, 'p': 0.19784972732128397, 'r': 0.5283828604103089}, 'rouge-2': {'f': 0.08256640287812875, 'p': 0.09299325006469149, 'r': 0.25628579443788835}, 'rouge-l': {'f': 0.12269860105541906, 'p':
threshold 0.3
Count: 256
Ref: президент украины владимир зеленский понизил возраст призыва в армию . так , если раньше повестки получали мужчины от 20 до 27 лет , то теперь призывать будут с 18 лет . при этом ранее украинские депутаты увеличили срок обучени
Hyp: украинские власти изменили призывной возраст , понизив его на два года . теперь призывать в вооруженные силы страны будут не с 20 лет , а с 18 . это следует из указа президента украины владимира зеленского . документ был опубли
BLEU:  0.08220594132948109
ROUGE:  {'rouge-1': {'f': 0.16309211337981533, 'p': 0.1219055996576763, 'r': 0.618088493205587}, 'rouge-2': {'f': 0.06353968800416003, 'p': 0.05008617358731043, 'r': 0.2990877206472207}, 'rouge-l': {'f': 0.09397859876267561, 'p': 0.
threshold 0.35
Count: 256
Ref: президент украины владимир зеленский не смог выполнить три своих предвыборных обещания . новый глава государства « не начал сажать коррупционеров » , не воплотил принцип « закон един для всех » и не вернул украинцев из эмиграци
Hyp: лидер платформы « национальная стратегия развития » дмитрий спивак заявил , что президент украины владимир зеленский не выполнил три предвыборных обещания . слова спивака передает телеканал newsone . он подчеркнул , что зеленск
BLEU:  0.08077665239427807
ROUGE:  {'rouge-1': {'f': 0.15237503586696544, 'p': 0.0929802965109001, 'r': 0.6494014132449097}, 'rouge-2': {'f': 0.05593057952165064, 'p': 0.03378298769683158, 'r': 0.3143778664364782}, 'rouge-l': {'f': 0.0832952159303457, 'p': 0.
threshold 0.4
Count: 256
Ref: парламент румынии вынес вотум недоверия правительству страны . депутаты обвинили премьера людовика орбана в « приватизации румынской демократии » всего через три месяца после его назначения . за отставку кабмина выступили более
Hyp: парламент румынии проголосовал за вынесение вотума недоверия правительству премьер-министра людовика орбана от национал-либеральной партии ( нлп ) . данное решение в ходе тайного голосования одобрил 261 из 465 депутатов . согла
BLEU:  0.08026268332794204
```

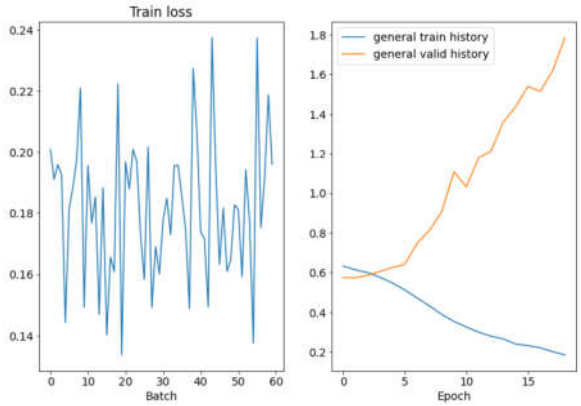
Обучим модель с весами и оценим ее качество

```
model = SentenceTaggerRNN(vocab_size, use_novelty=False).to(device)

train_all(True, N_EPOCHS, CLIP)

model.load_state_dict(torch.load('best-val-model.pt'))

top_k_inference_summarunner(model, test_iterator, 1)
top_k_inference_summarunner(model, test_iterator, 2)
top_k_inference_summarunner(model, test_iterator, 3)
top_k_inference_summarunner(model, test_iterator, 5)
threshold_inference_summarunner(model, test_iterator, 0.2)
threshold_inference_summarunner(model, test_iterator, 0.3)
threshold_inference_summarunner(model, test_iterator, 0.35)
threshold_inference_summarunner(model, test_iterator, 0.4)
threshold_inference_summarunner(model, test_iterator, 0.5)
```




```
Epoch: 20 | Time: 0m 8s
	Train Loss: 0.180 | Train PPL:   1.198
	 Val. Loss: 1.905 |  Val. PPL:   6.720
top_k= 1
Count: 256
Ref: неандертальцы хоронили собратьев , соблюдая определенные ритуалы , считают ученые . новая находка позволяет предположить , что неандертальцы создавали подобия могил , а более ранние данные указывают на то , что , возможно , при
Hyp: неандертальцы хоронили мертвецов с особыми ритуалами , считает команда исследователей под руководством специалистов из кембриджского университета .
BLEU:  0.1974612047722521
ROUGE:  {'rouge-1': {'f': 0.2816651701712057, 'p': 0.44265913216514763, 'r': 0.21911648617808385}, 'rouge-2': {'f': 0.14088060542198771, 'p': 0.22634890714244005, 'r': 0.10919055716826959}, 'rouge-l': {'f': 0.2109489572734253, 'p':
top_k= 2
Count: 256
Ref: убийство иранского генерала сулеймани , барическая пила от вильфанда , поражение сборной россии в финале молодежного чемпионата мира по хоккею — что интересовало телеграм-каналы на прошлой неделе .
Hyp: начнем с главного события новогодних дней — убийства американцами иранского генерала сулеймани иа « стекломой » называет его культовой фигурой шиитского мира . « с его деятельностью связывают практически всю « силовую » политик
BLEU:  0.4313127726205193
ROUGE:  {'rouge-1': {'f': 0.3087075074591505, 'p': 0.34382112511256513, 'r': 0.2977321270053813}, 'rouge-2': {'f': 0.14450677226244504, 'p': 0.1593672053278618, 'r': 0.1431857683675643}, 'rouge-l': {'f': 0.2610373441648967, 'p': 0.3
top_k= 3
Count: 256
Ref: президент россии владимир путин одобрил идею создания парламентского измерения « нормандского формата » . этот план он обсудил на встрече с лидером украинской партии « оппозиционная платформа — за жизнь » виктором медведчуком .
Hyp: президент россии владимир путин поддержал идею создать парламентское измерение « нормандского формата » с участием представителей россии , украины , германии и франции . « конечно , если парламентарии на украине , в россии , в
```

Обучим модель с novelty и без весов и оценим ее качество

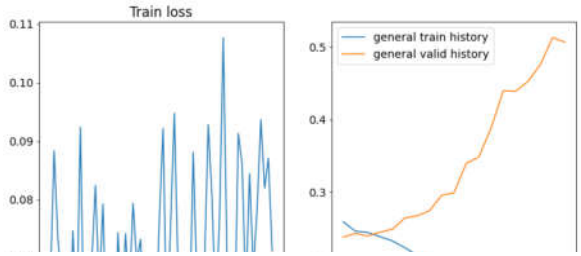
```
Ref: репутация принца гарри и меган маркл , вопреки стараниям нанятых голливудских пиарщиков , снова подорвана , на этот раз исправить имидж им помешал принц чарльз , выпустивший образцовый документальный фильм о своей работе
```

```
model = SentenceTaggerRNN(vocab_size, use_novelty=True).to(device)

train_all(False, N_EPOCHS, CLIP)

model.load_state_dict(torch.load('best-val-model.pt'))

top_k_inference_summarunner(model, test_iterator, 1)
top_k_inference_summarunner(model, test_iterator, 2)
top_k_inference_summarunner(model, test_iterator, 3)
top_k_inference_summarunner(model, test_iterator, 5)
threshold_inference_summarunner(model, test_iterator, 0.2)
threshold_inference_summarunner(model, test_iterator, 0.3)
threshold_inference_summarunner(model, test_iterator, 0.35)
threshold_inference_summarunner(model, test_iterator, 0.4)
threshold_inference_summarunner(model, test_iterator, 0.5)
```



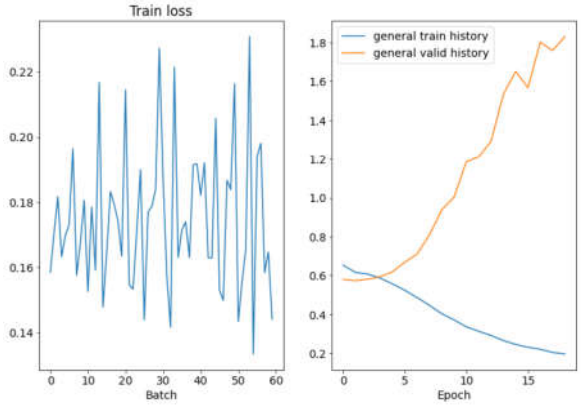
Обучим модель с novelty и весами и оценим ее качество

```
model = SentenceTaggerRNN(vocab_size, use_novelty=True).to(device)

train_all(True, N_EPOCHS, CLIP)

model.load_state_dict(torch.load('best-val-model.pt'))

top_k_inference_summarunner(model, test_iterator, 1)
top_k_inference_summarunner(model, test_iterator, 2)
top_k_inference_summarunner(model, test_iterator, 3)
top_k_inference_summarunner(model, test_iterator, 5)
threshold_inference_summarunner(model, test_iterator, 0.2)
threshold_inference_summarunner(model, test_iterator, 0.3)
threshold_inference_summarunner(model, test_iterator, 0.35)
threshold_inference_summarunner(model, test_iterator, 0.4)
threshold_inference_summarunner(model, test_iterator, 0.5)
```




```
Epoch: 20 | Time: 0m 9s
        Train Loss: 0.175 | Train PPL:   1.191
         Val. Loss: 2.318 |  Val. PPL:  10.160
top_k= 1
Count: 256
Ref: экипаж корабля « союз мс-16 » , который должен в апреле отправиться к мкс , заменят на дублеров , сообщили в роскосмосе . такое решение принято по медицинским показаниям , связанным с подготовкой на земле . неизвестно , сможет
Hyp: российская часть экипажа корабля « союз мс-16 » , который должен лететь к международной космической станции ( мкс ) 9 апреля , будет заменена на дублеров , сообщает « риа новости » со ссылкой на исполнительного директора роскос
BLEU:  0.19746212047722521
ROUGE:  {'rouge-1': {'f': 0.2816651701712057, 'p': 0.4426591321651474, 'r': 0.2191164861780839}, 'rouge-2': {'f': 0.14088060542198777, 'p': 0.22634890714244002, 'r': 0.10919055716826957}, 'rouge-l': {'f': 0.21094895727342552, 'p': 0
top_k= 2
Count: 256
Ref: в москве задержали следователя по особо важным делам , который получил взятку в размере 4,3 млн руб . по данным столичного главка ск , сотрудник правоохранительных органов получил деньги от родственника одного из обвиняемых , п
Hyp: в москве задержан следователь по особо важным делам — его обвиняют в получение взятки в 4,3 млн руб . ( часть 6 статьи 290 ук рф ) , сообщили в пресс-службе столичного главка ск . « 4 февраля следователь получил от лица , дейст
BLEU:  0.43145970664479413
ROUGE:  {'rouge-1': {'f': 0.3084082709165302, 'p': 0.34336524696611187, 'r': 0.29756229004885976}, 'rouge-2': {'f': 0.14441688650970821, 'p': 0.15926624164693487, 'r': 0.14310604897980922}, 'rouge-l': {'f': 0.26069235894158593, 'p':
top_k= 3
Count: 256
Ref: россия и белоруссия близки к согласованию « дорожных карт » по углублению экономической интеграции , констатировал глава минэкономразвития рф максим орешкин . но стороны признают , что остались нерешенными два ключевых вопроса
Hyp: москва и минск близки к завершению работы по согласованию « дорожных карт » по углублению экономической интеграции , заявил глава минэкономразвития рф максим орешкин в четверг в интервью телеканалу « россия-24 » . по его словам
BLEU:  0.4393225852198937
ROUGE:  {'rouge-1': {'f': 0.30652389358674786, 'p': 0.28833368357588474, 'r': 0.3471444940199836}, 'rouge-2': {'f': 0.13840721885461446, 'p': 0.1270066208602402, 'r': 0.164565228119747 05}, 'rouge-l': {'f': 0.2611095576131479, 'p': 0
top_k= 5
Count: 256
Ref: пелагея подтвердила расставание с хоккеистом иваном телегиным . в своем instagram певица рассказала , что не жалеет о трех годах семейной жизни , и выразила надежду на то , что они сумеют сохранить хорошие отношения ради их доч
Hyp: певица пелагея прокомментировала в своем instagram слухи о разводе с хоккеистом иваном телегиным . в комментарии к их совместному снимку она написала , что не жалеет о трех годах в браке , но заметила , что их история « закончи
BLEU:  0.32755461968439276
ROUGE:  {'rouge-1': {'f': 0.2909823295909378, 'p': 0.22947553661638934, 'r': 0.42051490536499797}, 'rouge-2': {'f': 0.12490268508054973, 'p': 0.09475886876642149, 'r': 0.19743851096766696}, 'rouge-l': {'f': 0.227945378421666, 'p': 0
threshold 0.2
Count: 256
Ref: нато готовится к крупномасштабному военному конфликту , заявили в генштабе вс россии . там обратили внимание , что в последнее время альянс все активнее начал продвигать тезис о « российской военной угрозе » . вместе с тем моск
Hyp: североатлантический альянс целенаправленно готовится к крупномасштабному военному конфликту — об этом свидетельствуют сценарии интенсивных военных учений блока , заявил на брифинге перед военными атташе иностранных государств н
BLEU:  0.40648338600957823
ROUGE:  {'rouge-1': {'f': 0.3034297965776053, 'p': 0.2716166615712585, 'r': 0.3664272957190601}, 'rouge-2': {'f': 0.13536679872468418, 'p': 0.11803209826849136, 'r': 0.17310298135058516}, 'rouge-l': {'f': 0.2545950167746059, 'p': 0.
threshold 0.3
Count: 256
Ref: крым по-прежнему ждет президента турции реджепа тайипа эрдогана в гости , чтобы развеять ошибочное представление о якобы притеснении крымских татар и об общей ситуации на полуострове — подчеркнул пресс-секретарь президента росс
Hyp: слова президента турции реджепа тайипа эрдогана о якобы « аннексии » крыма и плохом положении крымских татар не имеют реальных оснований , в чем он лично может приехать и убедиться , заявил пресс-секретарь президента россии дми
BLEU:  0.4405646302278032
ROUGE:  {'rouge-1': {'f': 0.30913350943014023, 'p': 0.2919557553565942, 'r': 0.34823454566798756}, 'rouge-2': {'f': 0.14076166353384603, 'p': 0.1304264136708097, 'r': 0.16506397535298628}, 'rouge-l': {'f': 0.2636397191434172, 'p': 0
threshold 0.35
Count: 256
Ref: мид японии потребовал от россии вернуть пять задержанных рыболовецких судов . траулеры из хоккайдо ранее были задержаны во время ловли осьминога у курильских островов . российские пограничники осмотрели восемь судов , три из ни
Hyp: министерство иностранных дел японии направило российским властям требование о возвращении пяти задержанных рыболовецких судов , пишет агентство kyodo . нота была передана москве по дипломатическим каналам , отметили в японском
BLEU:  0.4562767795084694
ROUGE:  {'rouge-1': {'f': 0.31102890706679376, 'p': 0.3029692056681351, 'r': 0.34047420640560394}, 'rouge-2': {'f': 0.14288364684877 79, 'p': 0.13636642905281002, 'r': 0.1624735214688 78}, 'rouge-l': {'f': 0.26542806891994486, 'p': 0.
threshold 0.4
Count: 256
Ref: в голландии в рамках борьбы с пиратством были отключены сервера плеера moonwalk . из-за этого россияне лишились доступа к крупным сайтам , которые нелегально распространяли фильмы и сериалы . между тем , статистика утверждает ,
Hyp: сервера плеера moonwalk , используемого в 80 % российских пиратских стриминговых сайтах , были отключены в голландии , передает портал torrentfreak . сообщается , что это решение остановило работу таких популярных ресурсов , ка
BLEU:  0.47906796342274143
ROUGE:  {'rouge-1': {'f': 0.31115174527853245, 'p': 0.31926216669700846, 'r': 0.3234887567004454 7}, 'rouge-2': {'f': 0.14357369501882952, 'p': 0.14573290470486452, 'r': 0.1538811785635389}, 'rouge-l': {'f': 0.2644912128784558, 'p':
threshold 0.5
Count: 256
Ref: « ред булл » собирался объявить состав пилотов на новый сезон на гран-при сша , но пока непонятно , выполнит ли это обещание . между этапами в мексике и на « трассе америк » консультант команды гельмут марко заявил о том , что
Hyp: уже когда даниил квят перед нынешним сезоном вернулся в « формулу-1 » , да еще и в программу « ред булл » , откуда чуть больше чем за год до возвращения был отчислен с фактической формулировкой « бесперспективный » , это выгляд
BLEU:  0.4329009527413664
ROUGE:  {'rouge-1': {'f': 0.30825257185970656, 'p': 0.34250757752045363, 'r': 0.2968641810735257}, 'rouge-2': {'f': 0.1438113415175967, 'p': 0.1584991781130881, 'r': 0.14169474144301206}, 'rouge-l': {'f': 0.2610835544402882, 'p': 0.
```

Вывод:

- использование весов улучшает результаты, когда используем threshold-based inference_summarunner
- использование novelty не улучшает результат
- максимальное значение BLEU, которого удалось достичь - это 0.47