Yet Another Conditional Headline Generation (Yachg)

Valetov D.K., Butov R.A.

May 2020

Abstract

Large-scale language models show promising results in text generation tasks. Several methods on conditional generation exists. We release another one, based on classic transformer with aim to enable text generating control via adding to source sequence an aspect vector produced by unsupervised aspect model. Code can be assessed in repository: `https://github.com/DmitriyValetov/nlp_course_project`.

1 Introduction

More the people, more texts they write, more time is spent in mining texts on related works, news, articles, less time for the nearest and dearest. Time spent on text surfing can be significantly reduced by good means of summarization, which will allow us not to dive into materials that do not correspond to our interests. Summarization is an important challenge of natural language understanding. The aim is to produce a shorten representation of an input text that captures the main meaning ideas of the original text. Aim of this paper is to hybridize topic modelling and summarization. Particularly - to use aspects vectors in the summary generation process and check whether an aspect can influence the result of summing text E.g. generate a different summary of the text by bias to one or more of its topics.

1.1 Team

Valetov Dmitriy: Data processing, implementing ABAE, conditioning a transformer with ABAE aspect vector.

Butov Roman: Data processing, metrics assessing, inference methods.

2 Related Work

Base of our problem is classic sequence-to-sequence problem, we need an instrument to generate target sequence with known source sequence. There are

two main types of summarization - extractive and abstractive. First extracts some key words from source sequence, and the second generates a fresh text extracting some latent information. Our problem is of the second type.

There are numerous works about text summation with recurrent neural networks [1, 2] and with transformers: [3, 4].

Works on conditional text generation similar to described in this paper could be found here: [5, 6]. The CTRL model is a language model that was trained to generate text with embedded tokens for conditioning the process like: "Horror lore", "Wikipedia article", "Books" etc.

3 Model Description

Base of our model is a classic transformer [7]. We used a classical implementation of transformer from Pytorch.

To make generation conditional we use sentiment extraction model (ABAE) from [8] to get most relative aspect vector for a given source sequence of tokens. Then we prepend this aspect vector to the given embedded source sequence (like it is done in [5]) and feed it in transformer. That's nearly all the model.

4 Dataset

Rossiya Segodnya dataset has been used [9]. We did several text normalizations: lemmatization with stopwords, BPE and raw text. First of all we did html parcing with BeautifulSoup package [10]. Sentence and word tokenization made by NLTK [11] package. For lemmatiztion we used pymorphy2 package [12]. Stopwords list obtained from NLTK package. For BPE we used youtokentome package [13].

Type	unique tokens	total tokens	vocabulary	tokens in vocabulary
lem+stop	600k	200M	50k	196M (98%)
bpe	50k	331M	50k	331M (100%)
raw	1.2M	262M	50k	244M (93%)

Table 1: Data normalization results

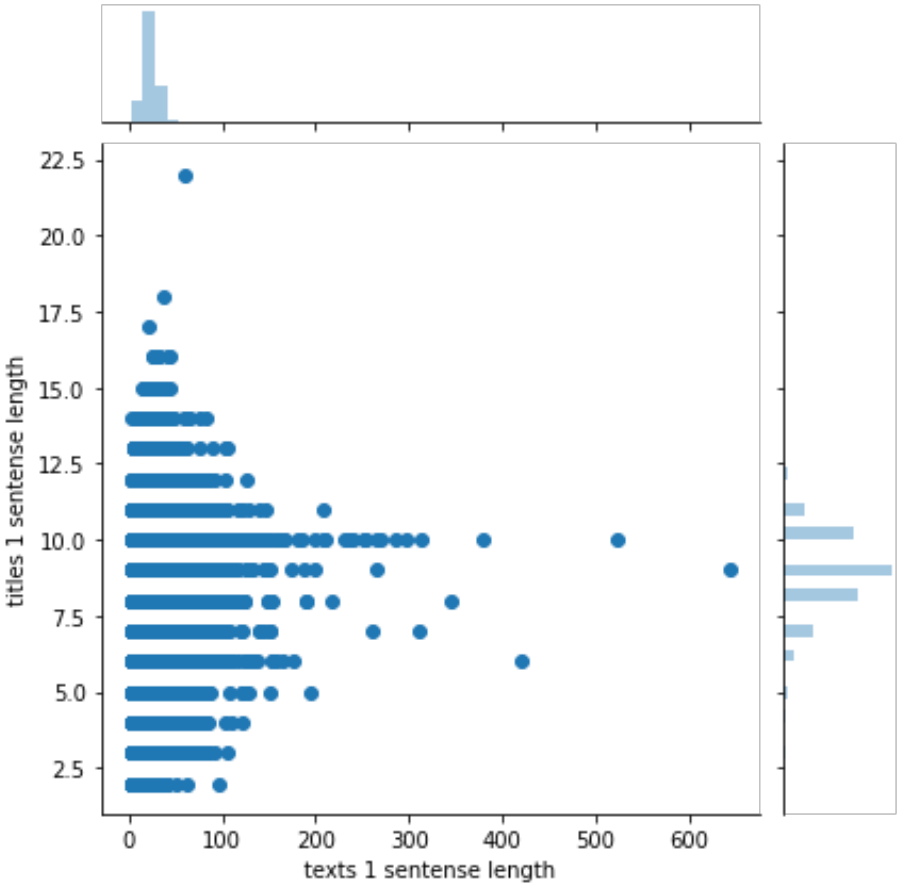

Figure 1: First sentences lengths distribution

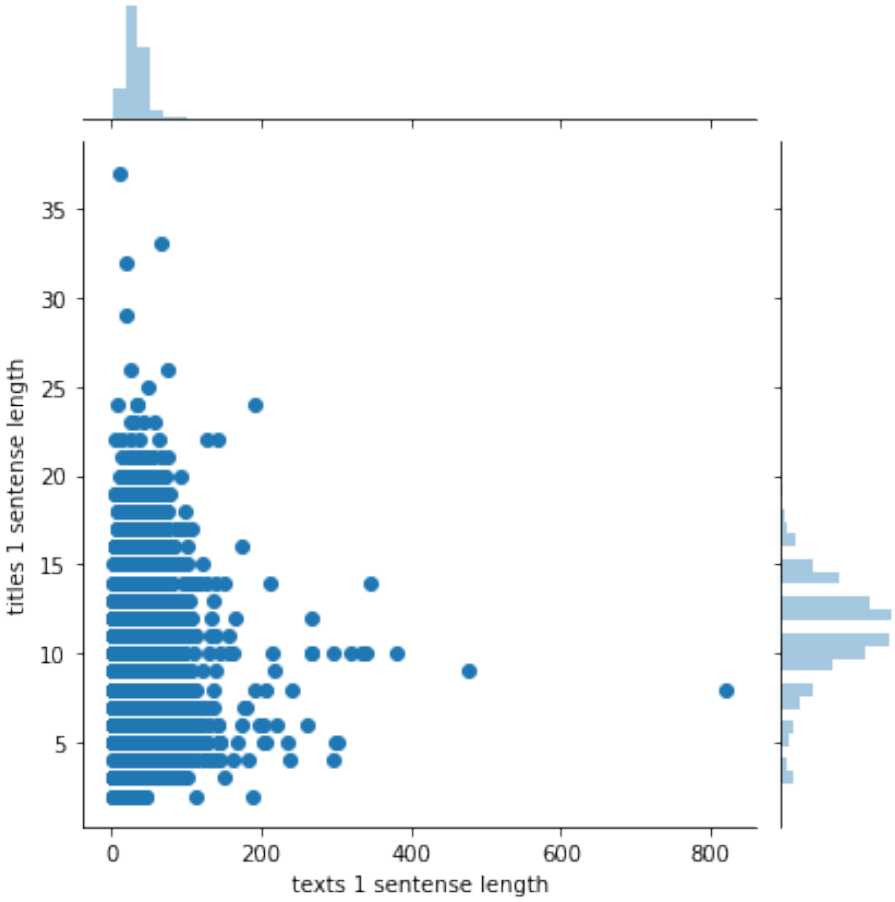

Figure 2: First sentences lengths distribution BPE

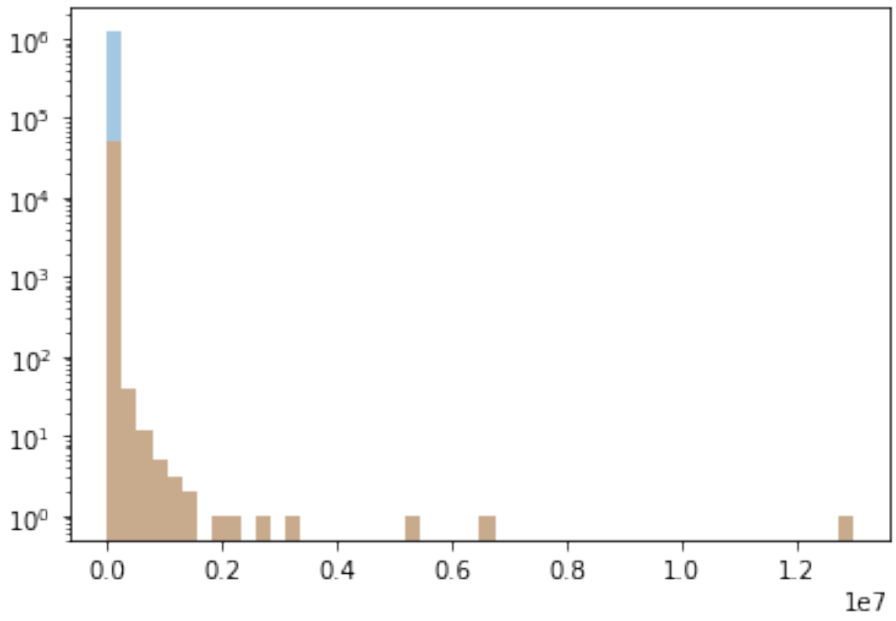

Figure 3: Raw word count distribution at reduced vocabulary (50k words, red) over full (1.2M words, blue)

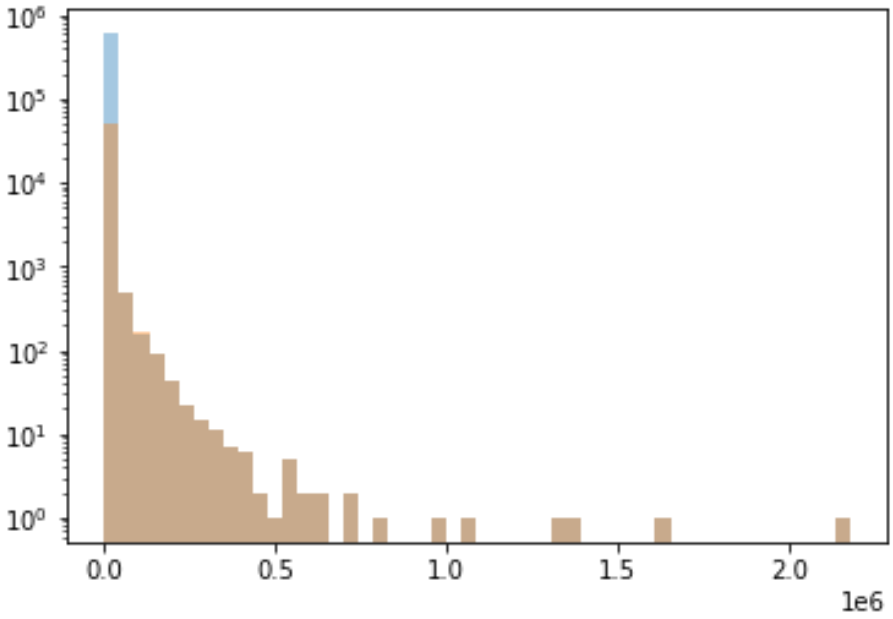

Figure 4: Lemmatized word count distribution at reduced vocabulary (50k words, red) over full (600k words, blue)

4.1 BPE

We tried to use BPE with the youtokentome [13] library. Conclusion is that for topics sharing the space with words (concept used in ABAE model) this approach doesn't seem to be much promising. Words may have several meanings, but tokens have much more and in fact have much more syntactic than semantic meanings. E.g. let's consider the attention matrix from RNN for word tokens: different words have strong relationships, but if we look at bpe matrix we rather find strong relationships between the same tokens than between different ones. But we have only qualitative analysis, for strict conclusions a quantitative analysis should be performed.

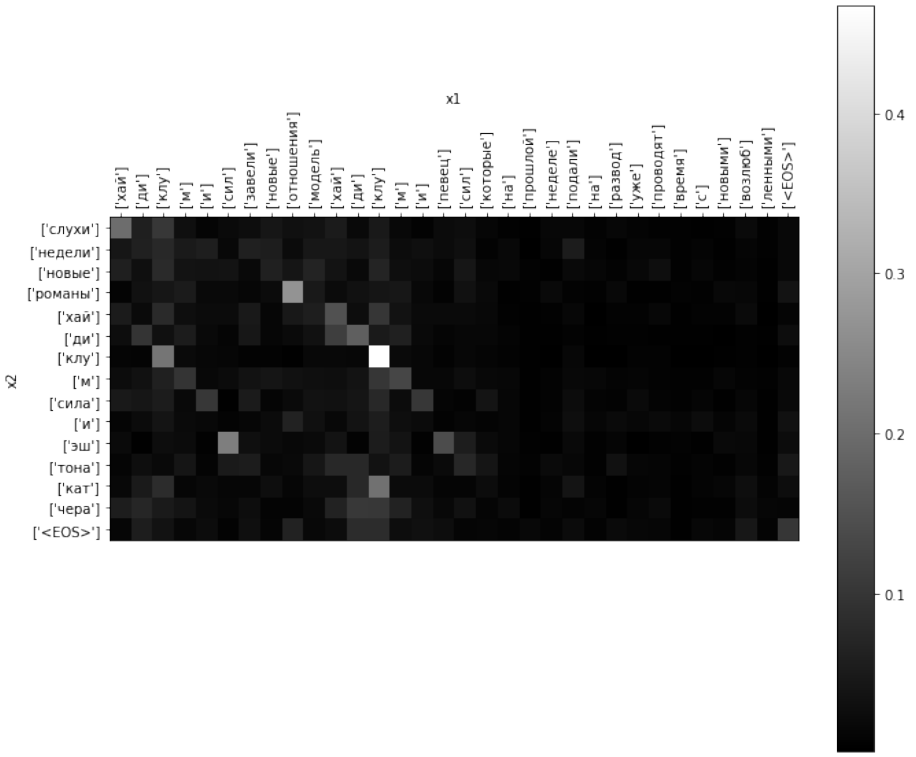

Figure 5: BPE attention martix

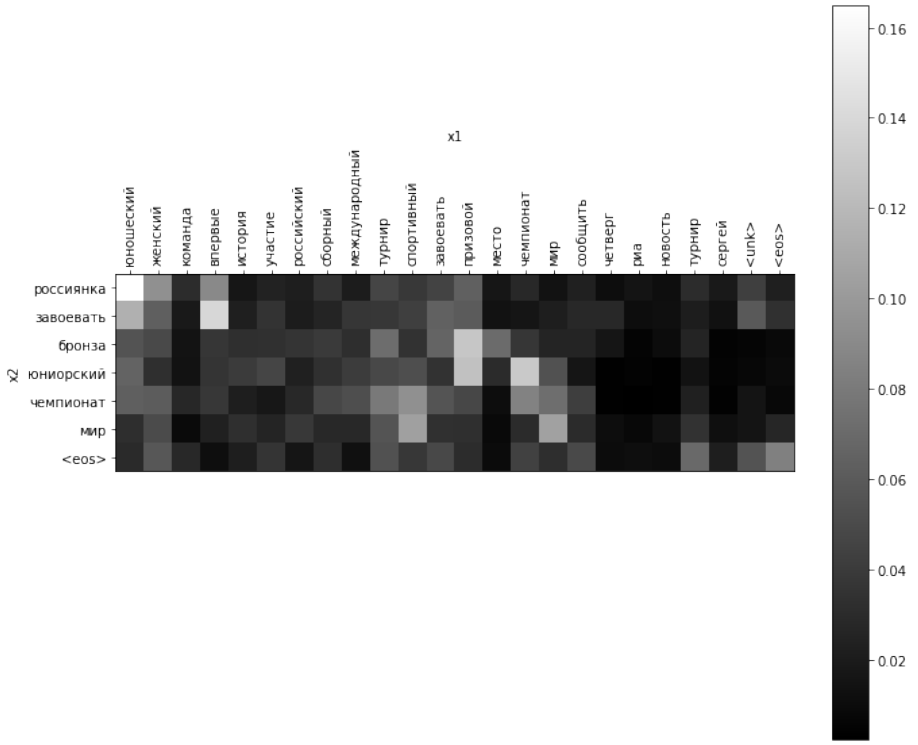

Figure 6: Words attention matrix

5 Experiments

5.1 Transformer + ABAE

Training process deserves detailed description. We pretrained word vectors by gensim word2vec model with the following parameters: window size is 10, negative sampling is 5, word embedding size is 300 as in all the way down through the transformer. Vocabulary is stricted by 50000 words, other words are encountered near 1-2 times through all corpus. ABAE aspects vectors are initiated with 100 kmeans centroids.

5.2 RNN

It is a seq2seq model with an attention mechanism. Principal scheme of the neural net is shown in Figure 7.

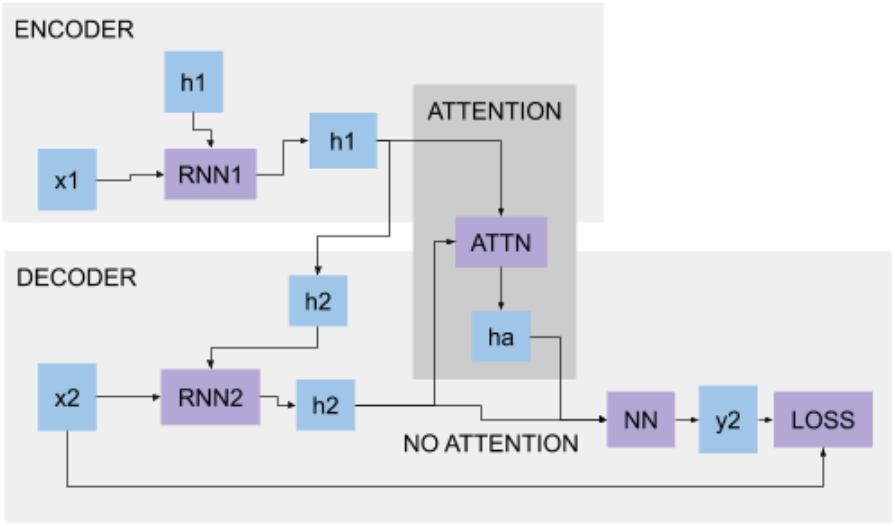

Figure 7: Seq2Seq RNN with attention scheme. Where: RNN - Recurrent Neural Network, NN - feedforward Neural Network with one hidden layer, x1 - input sequence, x2 - target sequence, y - prediction sequence, h - RNN hidden state, ATTN - attention algorithm

Parameters of the network are given in table 2. We used vanilla RNN cell types with a hidden size of 200. Output feedforward network NN makes conversion of sizes: 400 (200 RNN hidden + 200 RNN attention) -> 600 (NN hidden layer size) -> 50004 (embedding size).

Parameter	Value
Embedding dim	300
Embedding size	50004
RNN cell type	RNN
RNN hidden size	200
NN hidden size	600
Attention	softmin euclidean distance
Trained parameters	60M

Table 2: Model parameters

Figure 8: Attention algorithm comparison

We selected euclidean distance as an attention metric after comparison with dot, softmax dot, cos, softmax cos and inverse distance (dict distance metric is experimental and isn't considered in our task). Comparison of loss function after 5 epochs with several trials on a test dataset is shown in Figure 8. Loss with softmin distance function (i.e. softmax negative distance) is 10 times lower than with no attention model and 8 times lower than cos metric.

Parameter	Value
optimizer	Adam
learning rate	1e-3
weight decay	1e-6
batch size	300
seed	0

Table 3: Train parameters

5.3 Beam Search

We have implemented a beam search algorithm with several options, like backward search, batch and beam reductions.

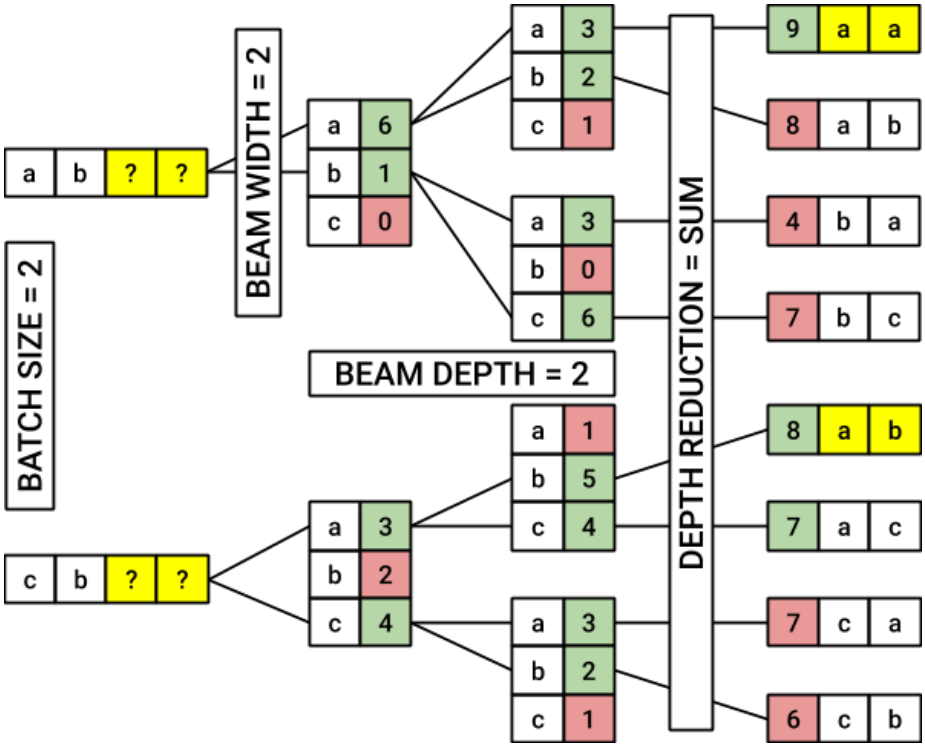

Figure 9: Classic beam search with depth reduction or cumulative score

Beam search implements the concept of chain of conditional probabilities between events. We use that sequence of events that have maximum likelihood over all steps. E.g. on a picture 9 given two sequences ab and cb for which we want to predict next tokens. These predictions have condition (previous tokens ab and c consequently) and weights for each possible token. E.g. for sequence cb weights for the next third token are 3 for token a, 2 for b and 4 for c. If we don't use beam search (one can say use beam search with beam depth 1), we choose more probable token c and go to prediction of fourth token with sequence cbc as new condition and so on. In our example we will have weigths 3, 2 and 1 for tokens a, b and c. In this case we would choose a fourth final token and produce a resulting sequence cbca. But what if we would choose a token on the first step? Then we would have another weights for a, b and c: 1, 5 and 4. Then we would choose b on for a fourth token and get the final sequence cbab. If we calculate the cumulative weight of each path we will see that the

last sequence has weight 8 that is more than weight 7 for the first one. We can argue about the meaning of this cumulative score or does it make sense. For probabilistic models its like a cumulative confidence about decision making. We can have small confidence on a first step but then strong on next that leads us to greater cumulative confidence. Beam search allows us to consider more paths. We can change beam width and depth. In a fact we can calculate all possible variants, there are N**D, where N - number of all possible tokens, D - number of predicting tokens. It is an exponential time task and requires a huge computational resources. In place of width we can stop beam if it achieves special termination token e.g. <eos>, but we can't know (for non exponential time) does it achieve it or doesn't. And put some restrictions on length or time for algorithm.

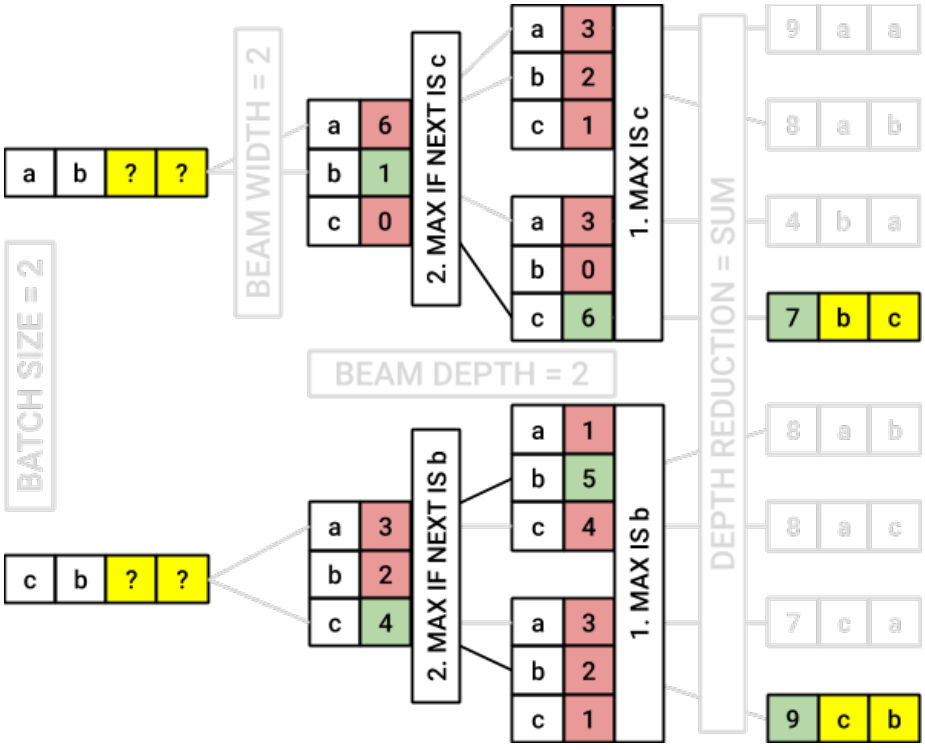

Figure 10: Backward algorithm

There is another variant how to calculate cumulative weight or rather choose right sequence. After calculating weights we can choose maximum weight token on a lst step and then go to previous step only in tokens that could produced max token. E.g. on picture 10 maximum token is b with weight 5. On previous step b can be achieved from a and c token. We choose c with weight 5 though

it doesn't lead to b directly.

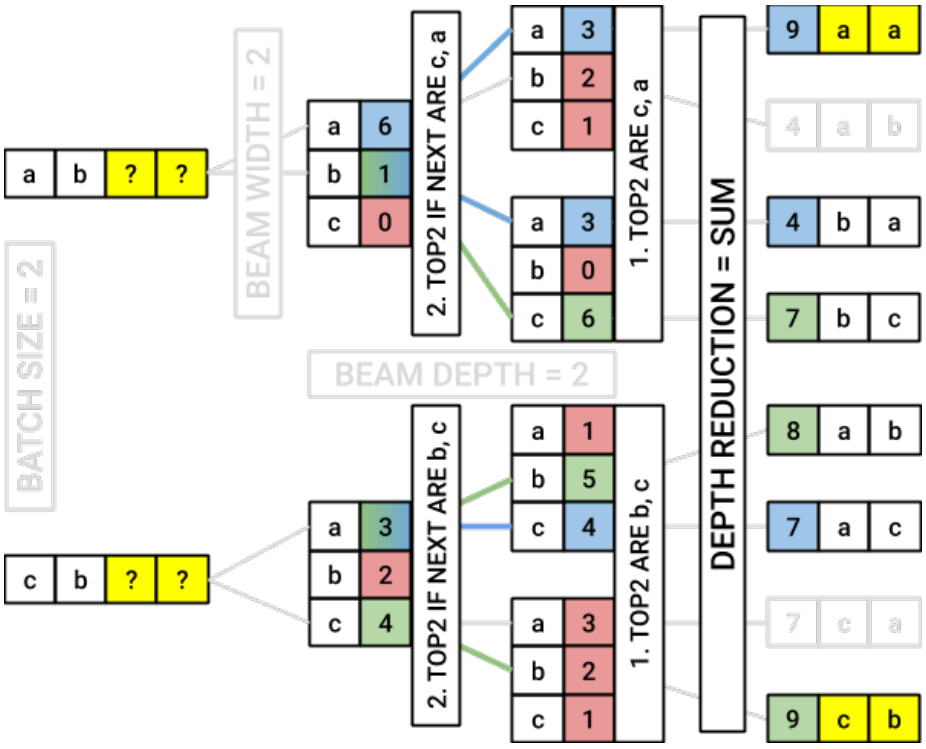

Figure 11: Backward algorithm (topk)

We can choose top k tokens at last prediction and propagate the algorithm independently.

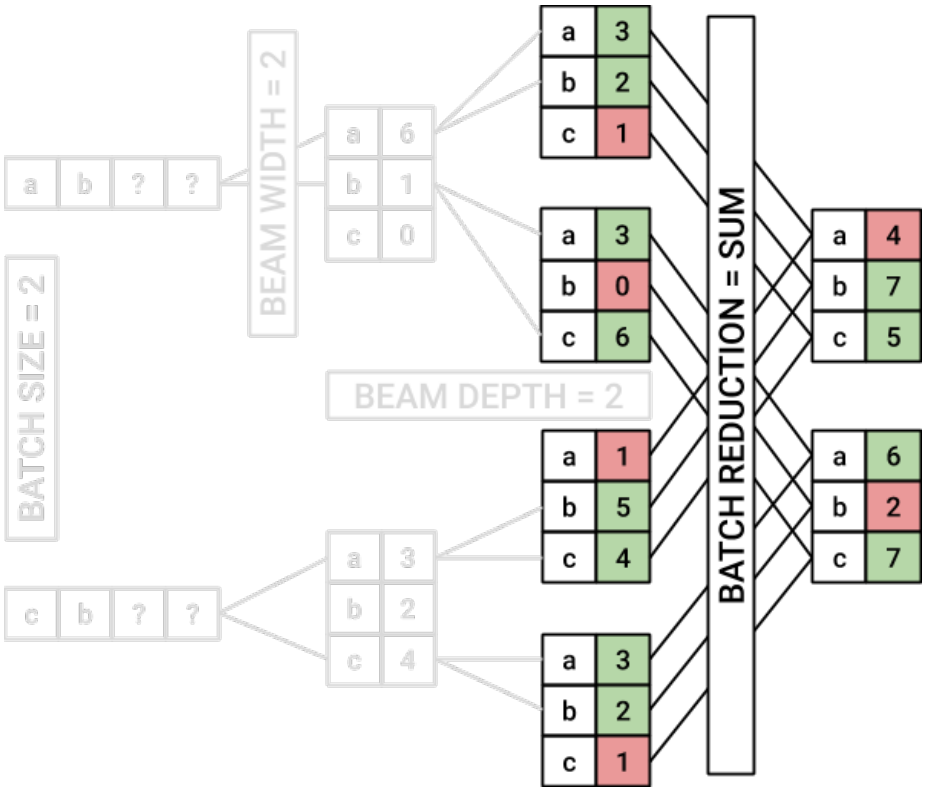

Figure 12: Batch reduction for many to one sequence generation

For topic generation where we have several sequences at text and one at topic we could try to do batch reductions to achieve cumulative batch likelihood for a token.

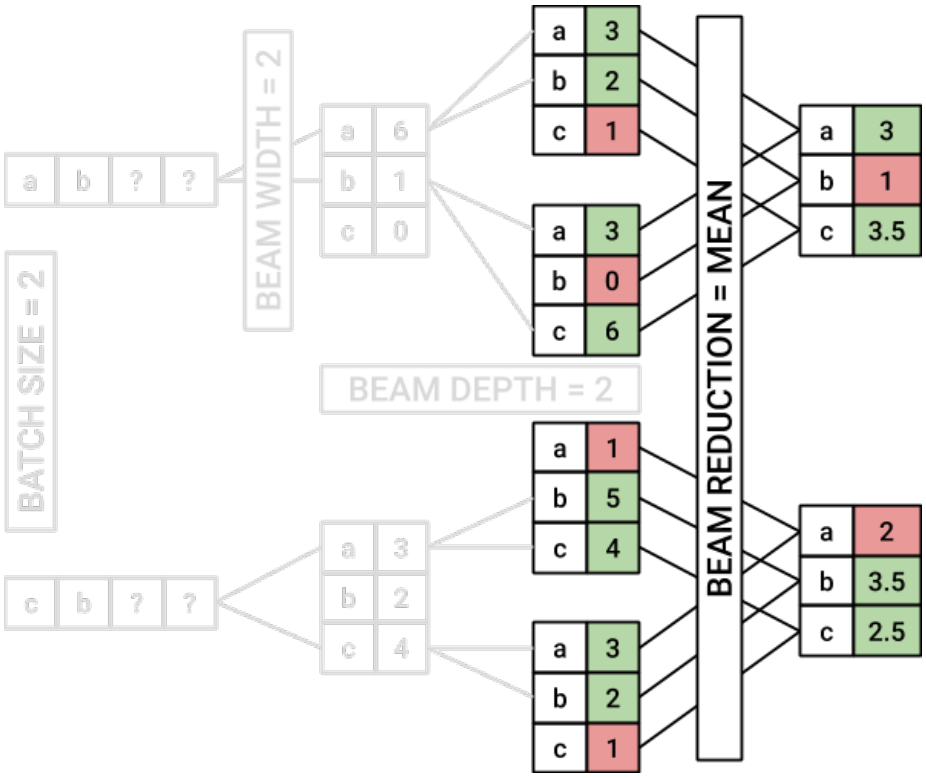

Figure 13: Beam reduction to share information between beams

If we have several beams we can make a beam reduction to achieve cumulative beams likelihoods for next propagation.

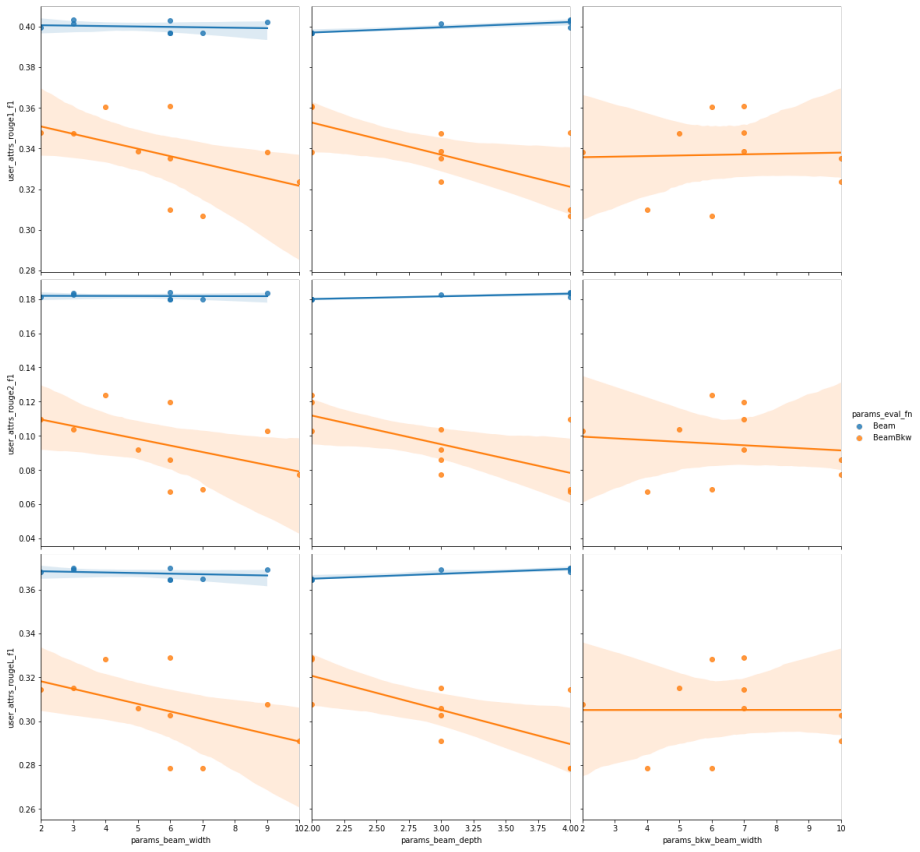

Figure 14: First sentences lengths distribution

From the comparison (Picture 14) we can conclude that forward-backward beam search has worse prediction than simple forward beam search. Also we see a small positive relationship between scores and beam depth and no relationship between beam width. Comparison made on 1000 samples from test dataset.

5.4 Metrics

We used ROUGE metrics to evaluate summarization quality. Also we measured length ratio and non ordered words coincidence (set).

5.5 Experiment Setup

Dataset was splitted into 3 parts: train, loss and validation with 70%, 20% and 10% of samples. We used only the first sentence for training, because that

reduces training time and there is a hypothesis that the topic of the news consists mostly of words of the first sentence. We used a GPU Tesla P100-PCIE-16GB.

5.6 Baselines

We got baseline results from papers [14] and [15].

6 Results

Scores comparison is given in the table 4. Scores obtained on 100k samples of the test dataset as mean of individual pairs scores. Outputs are given in the table 5.

model	tok	oov	input	R-1-f	R-1-r	R-2-f	R-2-r	R-L-f	R-L-r	R-m-f
T+A (greedy)	raw	7%	1S	12.66	12.60	2.10	2.11	12.25	12.21	9.00
RNN (greedy)	bpe	0%	1S	29.90	29.71	15.21	15.15	28.22	28.04	24.44
T (greedy)	lem	2%	1S	37.56	32.40	10.05	8.50	31.55	27.10	26.39
RNN (greedy)	raw	7%	1S	34.67	34.66	15.65	15.70	32.38	32.38	27.57
RNN (greedy)	lem	2%	1S	38.42	37.82	16.62	16.41	35.67	35.12	30.24
T+A (greedy)	lem	2%	1S	41.17	40.47	15.40	15.15	36.80	36.19	31.12
RNN (beam 10-10-3)	lem	2%	1S	39.60	39.14	17.97	17.78	36.84	36.42	31.47
RNN (beam 2-2-8)	lem	2%	1S	40.19	40.39	18.29	18.37	37.37	37.57	31.95
RNN (beam 10-2-10)	lem	2%	1S	40.17	40.58	18.49	18.66	37.34	37.72	32.00
First Sentence [14]	bpe	0%	1S	24.08	45.58	10.57	21.30	16.70	41.67	17.12
seq2seq-words-25m [15]	raw	?%	400T	36.96	35.19	19.68	19.02	34.30	33.60	30.31
UT w/ smoothing [14]	bpe	0%	3kT	39.31	37.10	21.82	20.66	36.32	35.37	32.48
Encoder-Decoder [14]	bpe	0%	1S	39.10	38.31	22.13	21.75	36.34	36.34	32.52
UT [14]	bpe	0%	3kT	39.75	37.62	22.15	21.04	36.81	35.91	32.90
seq2seq-bpe-25m [15]	bpe	0%	800T	40.30	38.83	22.94	22.18	37.50	37.01	33.58
copynet-bpe-43m [15]	bpe	0%	800T	41.61	40.33	24.46	23.76	38.85	38.51	34.97

Table 4: Rouge1-2-L-mean recall and f1 scores ordered by Rouge mean f1 score on RIA dataset. There tok - tokenization type, input - length of the input (S - sentence, T - token), oov - out of vocabulary words (from total words count), T - transformer, A - ABAE, UT - Universal Transformer. Beam [first token beam width]-[other tokens beam width]-[max beam depth].

Our models have worse score compared with related works. Also we got very low scores for Transformer with raw text normalization (Why?). Beam search raises scores on 1-2 points. We noticed that depth of the beam is more important than its width. We do not provide beam search results for Transformer because we didn't calculate them enough (100k samples). But on 1k samples it also gives 1-2 points to scores.

raw 1	
input	россия <unk> от арабского мира и международного сообщества не желая участвовать во встрече министров иностранных дел по сирии в париже заявил в четверг глава мид франции ален жюппе
target	россия <unk> не приехала на встречу по сирии мид франции
output	мид франции россия <unk> от арабского мира заявил ален жюппе
raw 2	
input	действия россии которая ввела эмбарго на импорт продовольственных товаров из ес сша и других западных стран принявших ранее санкции в отношении нее не стали сюрпризом их вероятность следовало просчитать озвучила в четверг комментарий президента латвии <unk> <unk> его лига <unk>
target	руководители латвии советуют производителям искать другие рынки сбыта
output	<unk> <unk> <unk> <unk> <unk> <unk>
lem 1	
input	консульский отдел американский посольство москва вторник днём перестать принимать посетитель ремонт сообщаться сайт посольство
target	консульский отдел посольство сша москва принимать посетитель
output	отдел посольство сша москва перестать принимать посетитель
lem 2	
input	читатель иностранный издание вступить дискуссия насколько закрытие проект южный поток расширение мощность трубопровод голуба поток повлиять благополучие страна евросоюз также дальнейший быть развиваться отношение россия турция
target	зарубежный пользователь отказ рф южный поток блестящий ход
output	читатель сми мочь южный проект поток снизить
bpe 1	
input	жители украины жалуются в координационный штаб общественной палаты россии на массовые нарушения своих прав заявил риа новости во вторник представитель оп
target	украинцы жалуются в общественную палату рф на массовые нарушения прав
output	жители украины жалуются на массовые нарушения прав человека
bpe 2	
input	лабораторное исследование выявило у четырех больных госпитализированных из кировского санатория колос с симптомами острой кишечной инфекции возбудитель сальмонеллеза сообщила в пятницу риа новости начальник отдела эпиднадзора управления роспотребнадзора по кировской области любовь опарина
target	отдыхающие санатория под кировом заболели сальмонеллезом
output	медики компенсацию поставили в санатории самарской области нехватки школьников

Table 5: Output samples

7 Conclusions

We used 2 methods for text summarization (RNN and Transformers), ABAE model for topic generation, Transformer + ABAE model for topic generation and summarization and several text normalizations algorithms. Conclusions and inferences:

1. We intended to implement Transformer + ABAE for text summarization and we've done it.
2. Transformer + ABAE model raised scores by several points in comparsion with simple Transformer.
3. Unfortunately we didn't try to change summarization topic by ABAE vectors due lack of time. In short: we need another dataset with tagged topics for this task.
4. We did several text preprocessing pipelines that led us to conclusion that BPE encoding is a very promising approach, that doesn't reduce dataset vocabulary, though it got worse scores on our models.
5. We implemented beam search with different options. We noticed that depth of beam search is more important than width though it leads to
6. We implemented simple seq2seq model with attention to compare it with transformer, and noticed that there is not much difference in scores between them.
7. Our scores is lower than scores in related works [14] [15]. There is several reasons from our point of view: our main task is to implement Transformer and ABAE model not to raise score of text summarization (but it's of course implied), we used self-made models and we did small hyperparameters tuning because of lack of time.
8. By analysing attentions matrices of RNN model we noticed that attention with BPE tokens seems more syntactic than semantic as one's with word tokens.
9. We implemented several beams reductions (like batch reduction or beam reduction) for beam search but didn't try them and will do so in future works.
10. Project turned out to be mostly research or even educational but this experience will be useful in our future projects.

References

[1] Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*, 2016.

[2] Piji Li, Wai Lam, Lidong Bing, and Zihao Wang. Deep recurrent generative decoder for abstractive text summarization. *arXiv preprint arXiv:1708.00625*, 2017.

[3] Xingxing Zhang, Furu Wei, and Ming Zhou. Hibert: Document level pre-training of hierarchical bidirectional transformers for document summarization. *arXiv preprint arXiv:1905.06566*, 2019.

[4] Andrew Hoang, Antoine Bosselut, Asli Celikyilmaz, and Yejin Choi. Efficient adaptation of pretrained transformers for abstractive summarization. *arXiv preprint arXiv:1906.00138*, 2019.

[5] Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*, 2019.

[6] Congying Xia, Chenwei Zhang, Hoang Nguyen, Jiawei Zhang, and Philip Yu. Cg-bert: Conditional text generation with bert for generalized few-shot intent detection. *arXiv preprint arXiv:2004.01881*, 2020.

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. pages 5998–6008, 2017.

[8] Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. An unsupervised neural attention model for aspect extraction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 388–397, 2017.

[9] Daniil Gavrilov, Pavel Kalaidin, and Valentin Malykh. Ria news Dataset. `https://github.com/RossiyaSegodnya/ria_news_dataset`, 2018. [Online; accessed 27-May-2020].

[10] Leonard Richardson. Beautiful soup documentation. *April*, 2007.

[11] Edward Loper and Steven Bird. Nltk: The natural language toolkit. 2002.

[12] Mikhail Korobov. Morphological analyzer and generator for russian and ukrainian languages. 542:320–332, 2015.

[13] Youtokentome - unsupervised text tokenizer focused on computational efficiency. `https://github.com/VKCOM/YouTokenToMe`.

[14] Daniil Gavrilov, Pavel Kalaidin, and Valentin Malykh. Self-attentive model for headline generation. In *Proceedings of the 41st European Conference on Information Retrieval*, 2019.

[15] Ilya Gusev. Importance of copying mechanism for news headline generation. *arXiv preprint arXiv:1904.11475*, 2019.