Document summarization on government's dataset with BART and PEGASUS

Revkov Sergey

May 2020

Abstract

Final work on the NLP course from Huawei. In this paper, study Bart and Pegasus, compile a new dataset and fine-tune both models on it. Link to project code right here: `https://github.com/GreySR/huawei-project`.

1 Introduction

The recent advances in multimedia and web-based applications have eased the accessibility to large collections of textual documents. To automate the process of document analysis, the research community has put relevant efforts into extracting short summaries of the document content. As the number of electronic text documents is increasing so is need for an automatic text summarizer. In contrast to extractive summarization which merely copies informative fragments from the input, abstractive summarization may generate novel words. A good abstractive summary covers principal information in the input and is linguistically fluent. In abstractive summarization, sequence-to-sequence has become a dominant framework using encoder-decoder architectures based on RNNs and more recently Transformers. In this paper i will try to summirize political or legal documents from here: `http://government.ru`. This documents are difficult to read and understand quickly. Therefore, the problem is not only technical in nature, but also causes difficulties in language modeling.

1.1 Team

Revkov Sergey @Greyss prepared this report, parsing and data collection, fine-tuning and evaluating dataset on BART and PEGASUS models.

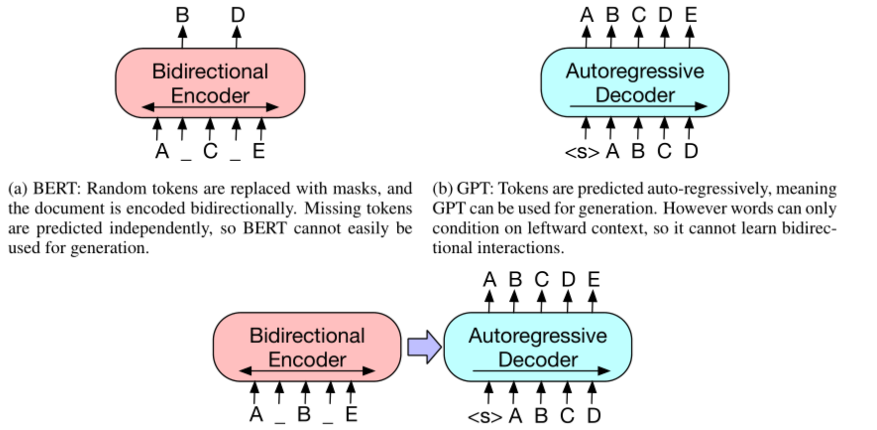
2 Related Work

BART [Lewis et al., 2019] which pre-trains a model combining Bidirectional and Auto-Regressive Transformers. BART is a denoising autoencoder built with a sequence-to-sequence model that is applicable to a very wide range of end tasks. Pretraining has two stages (1) text is corrupted with an arbitrary noising function, and (2) a sequence-to-sequence model is learned to reconstruct the original text. BART uses a standard Tranformer-based [Vaswani et al., 2017] neural machine translation architecture which, despite its simplicity, can be seen as generalizing BERT (due to the bidirectional encoder), GPT (with the left-to-right decoder), and many other more recent pretraining schemes (see Figure 1).

PEGASUS [Zhang et al., 2019] pre-training large Transformer-based encoder-decoder models on massive text corpora with anewself-supervisedobjective. In PEGASUS, important sentences are removed/masked from an input document and are generated together as one output sequence from the remaining sentences, similar to an extractive summary. Authors study pre-training objectives specifically for abstractive text summarization and evaluate on 12 downstream datasets spanning news, patents and legislative bills. They find that masking whole sentences from a document and generating these gap-sentences from the rest of the document works well as a pre-training objective for downstream summarization tasks. In particular, choosing putatively important sentences outperforms lead or randomly selected ones. Hypothesize this objective is suitable for abstractive summarization as it closely resembles the downstream task, encouraging whole-document understanding and summary-like generation. They call this self-supervised objective Gap Sentences Generation (GSG). Using GSG to pre-train a Transformer [Vaswani et al., 2017] encoder-decoder on large corpora of documents (Web and news articles) results in this method, Pre-training with Extracted Gap-sentences for Abstractive SUmmarization Sequence-to-sequence models, or PEGASUS. (see Figure 2).

3 Model Description

Because BART has an autoregressive decoder, it can be directly fine tuned for sequence generation tasks such as abstractive question answering and summarization. In both of these tasks, information is copied from the input but manipulated, which is closely related to the denoising pre-training objective. Here, the encoder input is the input sequence, and the decoder generates outputs autoregressively.
For fine-tuning PEGASUS was necessary to prepare the data in the required format {"inputs":tf.string, "targets":tf.string}.



(a) BERT: Random tokens are replaced with masks, and the document is encoded bidirectionally. Missing tokens are predicted independently, so BERT cannot easily be used for generation.

(b) GPT: Tokens are predicted auto-regressively, meaning GPT can be used for generation. However words can only condition on leftward context, so it cannot learn bidirectional interactions.

(c) BART: Inputs to the encoder need not be aligned with decoder outputs, allowing arbitary noise transformations. Here, a document has been corrupted by replacing spans of text with mask symbols. The corrupted document (left) is encoded with a bidirectional model, and then the likelihood of the original document (right) is calculated with an autoregressive decoder. For fine-tuning, an uncorrupted document is input to both the encoder and decoder, and we use representations from the final hidden state of the decoder.

Figure 1: Bart.

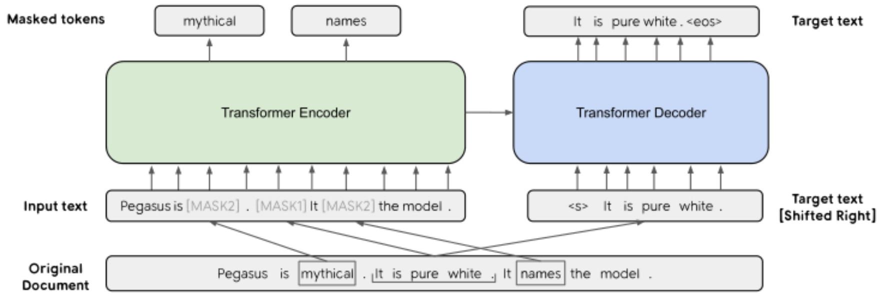
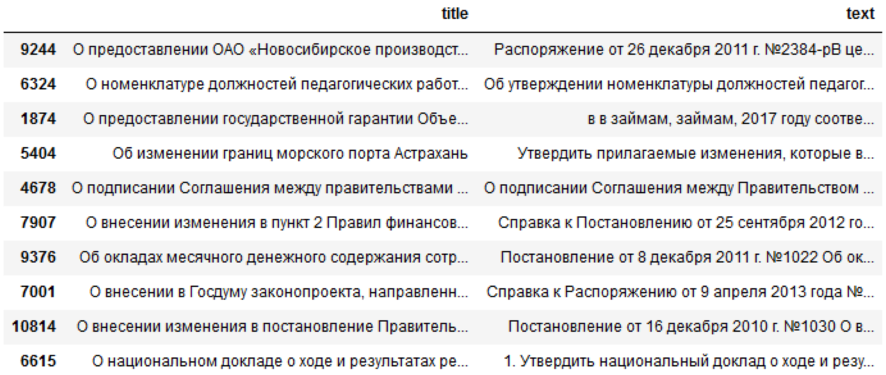


Figure 2: Pegasus.

4 Dataset

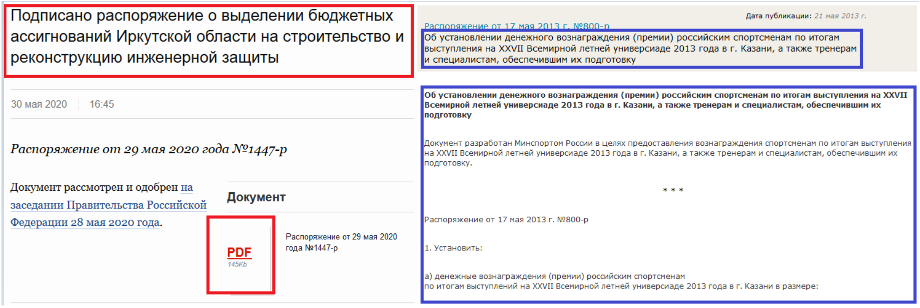
	Train	Valid	Test
Texts and Titles	8804	1101	1101

The dataset was compiled from the website of the government of the Russian Federation - http://government.ru/docs/ and http://archive.government.ru/gov/. For parsing used Python3 + BeautifulSoup. From the first site were downloaded docs in pdf's format. From other site extracted pure text from tag html. Both variants on Figure 4. For BART data represents in source and target



	title	text
9244	О предоставлении ОАО «Новосибирское производст...	Распоряжение от 26 декабря 2011 г. №2384-рВ це...
6324	О номенклатуре должностей педагогических работ...	Об утверждении номенклатуры должностей педагог...
1874	О предоставлении государственной гарантии Объе...	в в займам, займам, 2017 году соотве...
5404	Об изменении границ морского порта Астрахань	Утвердить прилагаемые изменения, которые в...
4678	О подписании Соглашения между правительствами ...	О подписании Соглашения между Правительством ...
7907	О внесении изменения в пункт 2 Правил финансов...	Справка к Постановлению от 25 сентября 2012 го...
9376	Об окладах месячного денежного содержания сотр...	Постановление от 8 декабря 2011 г. №1022 Об ок...
7001	О внесении в Госдуму законопроекта, направленн...	Справка к Распоряжению от 9 апреля 2013 года №...
10814	О внесении изменения в постановление Правитель...	Постановление от 16 декабря 2010 г. №1030 О в...
6615	О национальном докладе о ходе и результатах ре...	1. Утвердить национальный доклад о ходе и резу...

Figure 3: Ex. dataset.



Подписано распоряжение о выделении бюджетных ассигнований Иркутской области на строительство и реконструкцию инженерной защиты

30 мая 2020 | 16:45

Распоряжение от 29 мая 2020 года №1447-р

Документ рассмотрен и одобрен на заседании Правительства Российской Федерации 28 мая 2020 года.

Документ

PDF 145Kb

Распоряжение от 29 мая 2020 года №1447-р

Дата публикации: *21 мая 2013 г.*

Распоряжение от 17 мая 2013 г. №800-р

Об установлении денежного вознаграждения (премии) российским спортсменам по итогам выступления на XXVII Всемирной летней универсиаде 2013 года в г. Казани, а также тренерам и специалистам, обеспечившим их подготовку

Об установлении денежного вознаграждения (премии) российским спортсменам по итогам выступления на XXVII Всемирной летней универсиаде 2013 года в г. Казани, а также тренерам и специалистам, обеспечившим их подготовку

Документ разработан Минспортом России в целях предоставления вознаграждения спортсменам по итогам выступления на XXVII Всемирной летней универсиаде 2013 года в г. Казани, а также тренерам и специалистам, обеспечившим их подготовку.

* * *

Распоряжение от 17 мая 2013 г. №800-р

1. Установить:

а) денежные вознаграждения (премии) российским спортсменам
по итогам выступлений на XXVII Всемирной летней универсиаде 2013 года в г. Казани в размере:

Figure 4: Raw data.

files. For PEGASUS were prepared files in Tfrecords format - "inputs":tf.string, "targets":tf.string. More details in https://github.com/GreySR/huawei-project dataset.

5 Experiments

5.1 Metrics

ROUGE [Lin, 2004] stands for Recall-Oriented Understudy for Gisting Evaluation. It includes measures to automatically determine the quality of a summary by comparing it to other (ideal) summaries created by humans. The measures count the number of overlapping units such as n-gram, word sequences, and word pairs between the computer-generated summary to be evaluated and the ideal summaries created by humans. Formally, ROUGE-N is an n-gram recall between a candidate summary and a set of reference summaries. ROUGE-N is computed as follows Figure 5.

ROUGE-1 refers to the overlap of unigram (each word) between the system

$$\text{ROUGE-N} = \frac{\sum\limits_{S\in\{ReferemceSummaries\}} \sum\limits_{gram_n \in S} Count_{match}(gram_n)}{\sum\limits_{S\in\{ReferenceSummaries\}} \sum\limits_{gram_n \in S} Count(gram_n)}$$

Figure 5: ROUGE-N.

and reference summaries. **ROUGE-2** refers to the overlap of bigrams between the system and reference summaries. **ROUGE-L**: Longest Common Subsequence (LCS) based statistics. Longest common subsequence problem takes into account sentence level structure similarity naturally and identifies longest co-occurring in sequence n-grams automatically.

5.2 Experiment Setup

I fine-tuned BART and PEGASUS on my own dataset consisting 11010 titles and texts. Sentences for BART were encoded using byte-pair encoding [Sennrich et al., 2016], 0.8/0.1/0.1 - train/val/test. To help the model better fit the data, was disabled dropout for the final 10 % of training steps. Steps-20000, optimizer Adam, LR=3e-05, MAX_TOKENS=1024. Sentences for PEGASUS were encoded with SentencePiece Unigram algorithm (Unigram) [Kudo, 2018], 0.8/0.1/0.1 - train/val/test. Steps-20000, optimizer Adafactor, LR=0.0001, MAX_TOKENS=1024. Both models trained on Huawei Cloud with one GPU Tesla V100-PCIE-32GB.

5.3 Baselines

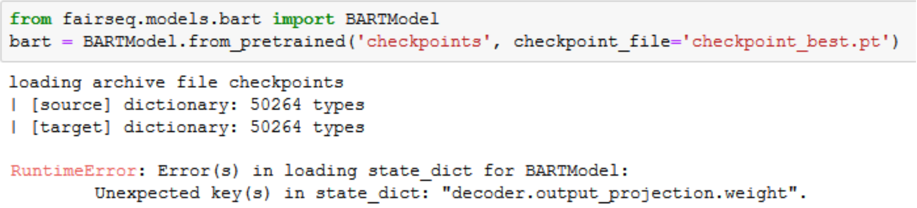
Results for large models BART and PEGASUS on CNN/DM datasets.

	R1	R2	RL
BART	44.16	21.28	40.90
PEGASUS	44.17	21.47	41.11

6 Results

Both models fine-tuned for approximately 10 hours each. Expand BART after training failed. I spent a lot of time preparing data, but it turned out to be more difficult. There are no such problems with data in English. Pegasus in turn loaded and evaluated but again, the problem with the Russian language and the encoding. Both results on Figure 6, 7.

	R1	R2	RL
PEGASUS	0.102515	0.042833	0.102450

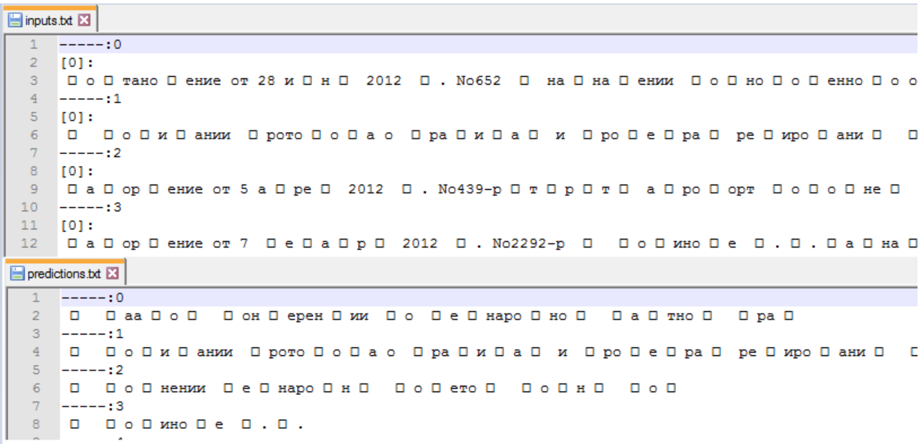


```
from fairseq.models.bart import BARTModel
bart = BARTModel.from_pretrained('checkpoints', checkpoint_file='checkpoint_best.pt')
```

```
loading archive file checkpoints
| [source] dictionary: 50264 types
| [target] dictionary: 50264 types

RuntimeError: Error(s) in loading state_dict for BARTModel:
	Unexpected key(s) in state_dict: "decoder.output_projection.weight".
```

Figure 6: Error - BART.



```
-----:0
[0]:
 □ о □ тано □ ение от 28 и □ н □  2012  □ . No652  □  на □ на □ ении  □ о □ но □ о □ енно □ о о
-----:1
[0]:
 □   □ о □ и □ ании  □ рото □ о □ а о  □ ра □ и □ а □  и  □ ро □ е □ ра □  ре □ иро □ ани □   □
-----:2
[0]:
 □ а □ ор □ ение от 5 а □ ре □  2012  □ . No439-р □ т □ р □ т □  а □ ро □ орт  □ о □ о □ не □
-----:3
[0]:
 □ а □ ор □ ение от 7  □ е □ а □ р □  2012  □ . No2292-р  □   □ о □ ино □ е  □ . □ . □ а □ на □
```

```
-----:0
 □   □ аа □ о □   □ он □ ерен □ ии  □ о  □ е □ наро □ но □   □ а □ тно □   □ ра □
-----:1
 □   □ о □ и □ ании  □ рото □ о □ а о  □ ра □ и □ а □  и  □ ро □ е □ ра □  ре □ иро □ ани □   □
-----:2
 □   □ о □ нении  □ е □ наро □ н □   □ о □ ето □   □ о □ н □   □ о □
-----:3
 □   □ о □ ино □ е  □ . □ .
```

Figure 7: Inputs and predictions - PEGASUS.

7 Conclusion

In this work, experience was gained in fine-tuning Bart and Pegasus. Identified problems with the dictionary of the Russian language. It is necessary to deal with BPE and other encoders. A unique dataset has been prepared, posted in the public domain.

References

[Kudo, 2018] Kudo, T. (2018). Subword regularization: Improving neural network translation models with multiple subword candidates.

[Lewis et al., 2019] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2019). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension.

[Lin, 2004] Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

[Sennrich et al., 2016] Sennrich, R., Haddow, B., and Birch, A. (2016). Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need.

[Zhang et al., 2019] Zhang, J., Zhao, Y., Saleh, M., and Liu, P. J. (2019). Pegasus: Pre-training with extracted gap-sentences for abstractive summarization.