News Summarizator

Alexander Kotyukov, German Abramov

May 2020

Abstract

The final project for Huawei Natural Language Processing course.
Link to our repo: `https://github.com/germanjke/huaweiNLP`.

1 Introduction

Text is the one of the most common types of information. At this point there is a need for computer linguistics methods that performs automatic data analysis in natural language. Among the NLP issues an important place is occupied by the task of automatic abstracting and summarizing the text. Manual writing of essays requires huge effort and automation of this process saves human resources.

Due to the fact that there are more and more news, including not only from traditional sources, but also from social networks, the task appears to somehow summarize everything that was said in the news. This will help a person decide whether to read this news. Thus, a person will be able to choose only interesting news for him and spend more time on one or another resource, which is what any company wants.

The prime task is trying to generate news header by it text, where original header can be used as target and original text used as train data. Success solution can be used for "shortization" long texts without loosing general line of story. We want to try and compare two types of summarization, extractive and abstractive.

This work presents an attempt to build a text summarization system using a neural network for the example of generating news headlines. The relevance of this work is determined by the need for the development of automatic summarization of text. Algorithms, methods for creating a news headline in particular, can be of interest to news agencies and search engines.

1.1 Team

Our team is:

Alexander Kotyukov (github @kotyukov) design parser, dataset collecting, summarization model, document preparation

German Abramov (github @germanjke) dataset collecting, summarization model, document preparation

2 Related Work

The prior idea of our work was research two main approaches to summarizing the text:

1) Extractive summarization. This method of summarization is based on the extraction of key phrases from source documents or texts and the combine them to the final text;

2) Abstractive summarization. This approach to summarization involves paraphrasing and shortening the source text. An abstract approach to summa1 rization allows the use of completely new phrases and sentences in the text of the abstract - exactly the same way as a person does.

Unfortunately, due to lack of time, we worked only with the first part of the task. So on, only extractive summarization methods will be considered in the report.

The following libraries for Python language were used in this work:

Pandas - a library for processing and analyzing data;

Numpy - a library that provides work with multidimensional arrays and matrices;

PyTorch - a library for building neural networks on Python language;

NLTK - a library for processing natural language;

Razdel - a library for word processing in Russian;

Rouge - a library for calculating the ROUGE metric.

We tried to get ideas from the article Self-Attentive Model for Headline Generation. It was especially helpful to learn about BPE [Gavrilov et al., 2019]. Another article that we using was Importance of copying mechanism for newsheadline generation [Gusev, 2019]. This arcticle helped us with first sentence extractor, selfdesigned TextRank and oracle summary. Finally, extractive RNN model was builded by this article [Nallapati et al., 2016]. We used simple models (first sentence dumb extractor (FSDE), Selfdesigned TextRank (and same with MorphAnalyzer), Oracle Summary) as a base line for our task. The main model for solving our problem is RNN.

3 Model Description

RNN, whose graphical representation is presented in Figure 1. The first layer of the RNN runs at the word level, and computes hidden state representations at each word position sequentially, based on the current word embeddings and the previous hidden state. We also use another RNN at the word level that runs backwards from the last word to the first, and we refer to the pair of forward and backward RNNs as a bidirectional RNN. The model also consists of a second layer of bi-directional RNN that runs at the sentence-level and accepts the average-pooled, concatenated hidden states of the bi-directional word-level RNNs as input. The hidden states of the second layer RNN encode the representations of the sentences in the document. The representation of the entire document is then modeled as a non-linear transformation of the average

pooling of the concatenated hidden states of the bi-directional sentence-level RNN [Nallapati et al., 2016].

In order to train our extractive model, we need ground truth in the form of sentence-level binary labels for each document, representing their membership in the summary. Our approach is based on the idea that the selected sentences from the document should be the ones that maximize the Rouge score with respect to gold summaries. Since it is computationally expensive to find a globally optimal subset of sentences that maximizes the Rouge score, we employ a greedy approach, where we add one sentence at a time incrementally to the summary, such that the Rouge score of the current set of selected sentences is maximized with respect to the entire gold summary . We stop when none of the remaining candidate sentences improves the Rouge score upon addition to the current summary set. We return this subset of sentences as the extractive ground-truth, which is used to train our RNN based sequence classifier [Nallapati et al., 2016].

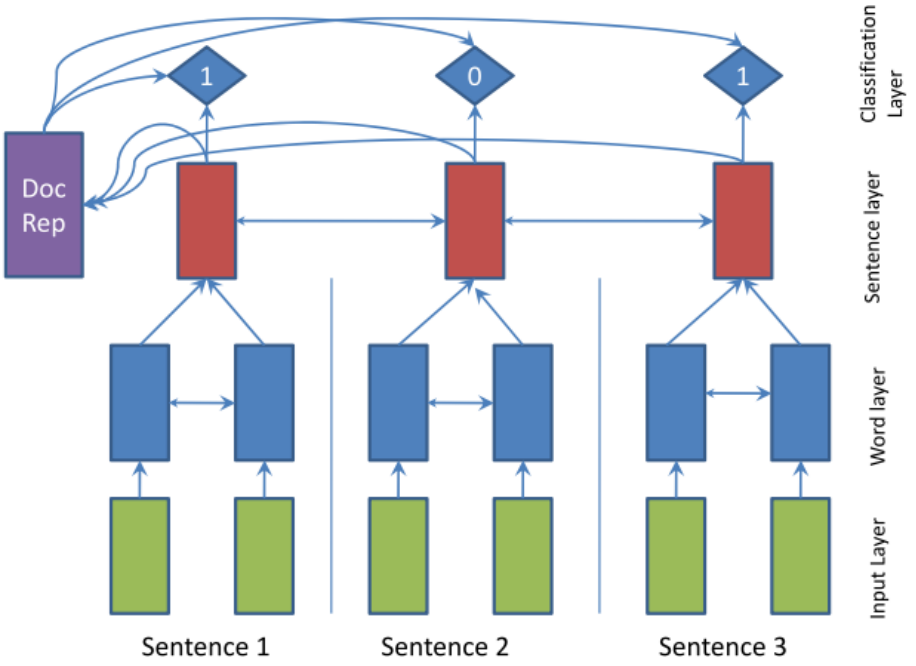

Figure 1: SummaRuNNer: A two-layer RNN based sequence classifier: the bottom layer operates at word level within each sentence, while the top layer runs over sentences. Double-pointed arrows indicate a bi-directional RNN. The top layer with 1's and 0's is the sigmoid activation based classification layer that decides whether or not each sentence belongs to the summary. The decision at each sentence depends on the content richness of the sentence, its salience with respect to the document, its novelty with respect to the accumulated summary representation and other positional features [Nallapati et al., 2016].

4 Dataset

We collected several news texts from website of Rossiyskaya Gazeta. Licens agreements permits collecting, copying and saving data from this site for personal non-commersial uses by Russian citizens. Metodology to collect the dataset is self-designed parser through C sharp and AngleSharp library. Raw HTML -> JSON dataset.

In our experiment, this dataset was used. It contains 134010 records of news texts along with their headers, source links, tags, authors, date and categories (for example, economics, sports, society, etc.). The news presented in the corpus mainly related to the period from August 2019 to April 2020. Due to specific of html and css formatting of some special news pages, some texts were collected with errors. Therefore, the dataset was cleaned from empty records, invalid entries containing only html tags was manually checked and dropped. At the data preprocessing stage, texts were cleaned using regular expressions, all words were reduced to lowercase. 133,773 entries were included in the final set. For the test sample, 12.5 percents of all data was given, the validation part was 15 percents of the training sample. Pre-trained embeddings was not used.

5 Experiments

Let's talk about metrics and our baselines

5.1 Metrics

One of the most common metrics to measure of sequences is considered the metric BLEU score (the Bilingual Evaluation Understudy).

This metric was originally created to evaluate machine translation. The principle of its work is that it compares the generated version of the text with target text (created by a person) and gives a value in the range from 0 to 1. If it matches completely, the BLUE score value is equal to one: there is no difference between the generated and the original text. If the similarities were not found, the BLUE score is zero.

It is important to note that in reality, when evaluating machine translation, the ideal BLUE score does not occur, because creating a translation that is completely identical to the original translation seems an impossible task even for a professional translator.

Comparison of the generated and the original sequence occurs by counting matches between n-grams in the generated texts and the original (target) texts. In this case, one token will be represented as a unigram, a pair of tokens as bigrams, etc. This comparison is made without attention to the word order. 2

Another metric that we used to measure sequences, is ROUGE (RecallOriented Understudy for Gisting Evaluation). Initially, this metric was created to analyze automatic text summarization. The ROUGE metric is based on a

Recall value that displays how much the generated text is intersect with the target text.

There are several varieties of ROUGE metrics what we used, they include the ROUGE-1 metric, which is based on the coincidence of the unigrams in the target and generated text. ROUGE-2 based on the coincidence of the bigrams, and ROUGE-L - the longest sequence of matching words.

5.2 Baseline

For baseline were taken fairly simple models that are able to compress the text. First model is FSDE aka first sentence dumb extractor. Our hypothesis is that in many news the first sentence of text already contains the most important summary. From the good literacy, that kind of duplication in the title looks silly, but as a baseline method we want to check what metrics it can achieve. Then, the functions of calculating the proximity of sentences based on word intersection, compiling a summary using TextRank, and also the greedy construction of oracle summary were set.

6 Results

The table shows the best results for all experiments

Model/Metric	BLEU	ROUGE-1	ROUGE-longest-r
Lead-1	0.1830	0.2528	0.2314
TextRank	0.1087	0.1925	0.1737
TextRank(w/MorphAnalyzer)	0.1088	0.2002	0.1807
Oracle Summary	0.1990	0.3243	0.3019
Extractive RNN	0.2023	0.2704	0.2471

During training process, the model reached the loss value of 0.049895 at the training set and 0.063467 at the validation set. For test dataset generated headers scoring with BLEU and ROUGE metrics, the best results were shown: RNN-model - BLEU score 0.2023, ROUGE-L-r score 0.2471

As a reference metric values, we use score values from the MIPT DoIHT NLP training notebook by Iliya Gusev (the gazeta.ru dataset was used): RNN-model - BLEU score 0.19034, ROUGE-L-r score 0.140083 TextRank - BLEU score 0.27914, ROUGE-L-r score 0.20339

References

[Gavrilov et al., 2019] Gavrilov, D., Kalaidin, P., and Malykh, V. (2019). Self-attentive model for headline generation. *arxiv.org, arXiv:1901.07786*.

[Gusev, 2019] Gusev, I. (2019). Importance of copying mechanism for news headline generation. *arxiv.org, arXiv:1904.11475*.

[Nallapati et al., 2016] Nallapati, R., Zhai, F., and Zhou, B. (2016). Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. *arxiv.org, arXiv:1611.04230*.