Headline generation: first sentence vs neural machine translation

Churikov N. S. (nikita@chur.ru),
Sannikova E. (elena.sannikova59@gmail.com)

Saint Petersburg State University, Saint Petersburg, Russia
Behavox, Saint Petersburg, Russia

In this article, we describe our experience of participating in a headline generation contest organized by VKontakte. We took the third place in the competition by modifying the baseline solution through the data cleaning. In addition, we tried to train and apply the transformer architecture combined with byte pair encoding, but this solution turned out to be worse than the baseline. At the end, we present our results on leaderboard for different solutions, and ROUGE scores on our test set.

Key words: text summarization, headline generation, Russian language

Генерация заголовков: первое предложение против глубокого машинного перевода

Чуриков Н. С. (nikita@chur.ru),
Санникова Е. (elena.sannikova59@gmail.com),

Санкт-Петербургский государственный Университет, Санкт-Петербург, Россия
Behavox, Санкт-Петербург, Россия

В данной статье мы описываем наш опыт участия в конкурсе по генерации заголовков, организованном ВКонтакте. Мы заняли третье место в соревновании модифицировав пороговое решение через очистку от лишней информации. Это позволило нам перевалить за решение-baseline. Помимо этого, мы попробовали обучить и применить трансформер токенизируя данные через byte pair encoding, однако это решение оказалось хуже порогового. В конце мы приводим численные результаты, полученные в соревновании, а также оценки, полученные на нашей тестовой выборке.

Ключевые слова: автореферирование текстов, генерация заголовков, русский язык

1 Introduction

The problem of generating headers is a specific summarization problem, where output of a model is just a short single sentence. Conceptually, there are two approaches to the problem: extractive and abstractive summarization. In the case of extractive summarization, only the original text is used. Abstractive summarization is different in that the output text may contain words that were not in the source material.

In this article, we describe our results in the header generation competition, which was organized by VKontakte for Dialog Evaluation. In our work, we used extractive and abstractive summarization. The methods used by us are described in the experiments section, the results of our models on the leaderboard and on our test set are given in the results section. At the end of the work, we present our conclusions on the competition and our thoughts on what can be improved.

2 System description

In the headline generation competition, we've decided to compare classical summarization approaches (textrank [1]), which do not require training and approaches based on neural networks. As a result, we've managed to get the results in the leaderboard only for part of our experiments, specifically for the first sentence, transformer-architecture [10] with BPE trained on Wikipedia [3] and trained on the competition data. We used the implementation of the transformer from the library of Open Neural Machine Translation [4]. The launch code for training, predictions, and models can be found in our repository [1].

3 Data and training

Competition data was provided in the form of raw pieces of original html pages from Ria News. This means that there were various html tags and entities present in the data. And each news and a matching headline in raw text.

```
<p> <strong> <\strong> <\p> \n <p> <strong> Moscow, Dec 1   &mdash;
RIA news. <\strong> a fire in   one of the   workshops in
 ...<\p>
```

As a result, the first thing we tried was to clean up the data from unnecessary information. So, we've made a preprocessor that removes all html tags and entities.

In addition, we've found that sometimes there is no text in the data, since the original news is represented by some form of image (for example, a screenshot of Twitter), and the news is purely Polish. These are all outliers we cleared data from.

However, if our models got empty string during inference, we used most common word in headlines which was "заявил".

Then, for training and validation of transformer, we've split the data in such a way that 90% of data goes for training and 10% for test. For test set creation we just shuffled data. For reproduction purposes we provide in our repository test set file.

As for byte pair encoding, BPE pretrained on Wikipedia turned all numbers into zeros. However BPE trained from scratch on RIA news dataset preserved numbers.

4 Experiments

For all experiments, we used the preprocessing as described above. For training of our models we used 8 Nvidia K80.

4.1 Evaluation

It is common practice to use ROUGE score [5] in summarization problems. In practice, so-called ROUGE 1,2, L - precision, recall, F1 metrics are being used. The names in ROUGE X - Y denote the following: X is the number of ngramm used to calculate the Y metric. In the case of ngramm of size L, the longest common subsequence from predicted sequence found in original sequence. Y metrics are classic accuracy, recall and F1 score.

In the competition, for the calculation of quality, the average of ROUGE 1, 2, L - F1 score was used.

[1] https://github.com/kuparez/headline_generator

4.2 First sentence

As described in [7], the first sentence for automatically creating a news headline is a very strong baseline. In particular, in the headline generation competition, the first sentence of news articles was proposed as a baseline solution.

However, we used our knowledge about the data and besides removal of html tags and entities we've also skipped first sentences with "риа новости" in it. This allowed us to skip sentences with information about the date the news was posted as these sentences did not contain any information besides that.

4.3 Textrank

We used the classic extractive summarization algorithm - textrank [1]. We took implementation of this algorithm from gensim [8] To create a summary using keywords and original sentences we've set extraction size to be 20% of the original text.

4.4 Transformer

Since the competition was held for only a month, we've decided to repeat the results described in the article VKontakte [2]. We took the implementation of the transformer from Open NMT, with the parameters described in the article.

Then, we tried to focus on what to use as input for the model. We tried the first sentence according to the rules above. We've also tried as an input first 2000 BPE tokens.

Due to restrictions on VK servers we could not make very complex predictions. There for, during predictions on leaderboard we've used beam search of size 5. But for evaluation on our test set we used beam size of 10.

5 Results

As could be seen in Table 1, none of the models we've trained could show better results than the preprocessed first sentence. Unfortunately, we could not check some of our models on VK servers due to the restrictions on time complexity of our models. For this reason, we also present results on our test set.

Algorithm	Score
Baseline	0.19500071
First sentence	0.19502427
Wiki BPE transformer (first sentence, no beam search=5)	0.16397515
Ria BPE transformer (2000 first tokens, beam search=5)	0.16131584
Textrank summarization	0.10764881
Textrank keywords	0.06259589

Table 1: Results from evaluation by VK servers. Score is mean of ROUGE-1,2,L F1 score.

Table 2 show results of evaluation on our test set. Surprisingly, we've got much better results on our split, than on public leaderboard. And specifically, neural networks perform much better than first sentence model. It's worth pointing out that transformer trained with RIA BPE perform slightly worse than transformer with Wiki BPE. We think, the reason for that might be that model is more robust, because it doesn't need to predict any variation of numbers, because all numeric data is reduced to zeros. Unfortunately, taxtrank based

model didn't show good results for F1 score, but recall is pretty high. It's probably due to extraction of too much information.

Another thing is that local evaluation show better results than VK server evaluation. So we decided to check for data leaks. We assumed that there could be none, because for traintest split we used scikit learn [6] train test split method. We checked, how many texts in our test set are identical to texts in train. It turned out that 975 texts in test set were found in train set.

Then we decided to check original data and find how many training examples are identical. We found, that 2651 texts are identical.

Algorithm\Score	1F	1P	1R	2F	2P	2R	LF	LP	LR
First sentence	0.23	0.16	0.44	0.10	0.07	0.21	0.16	0.15	0.40
Wiki BPE transformer (first sentence, beam=10)	0.37	0.39	0.36	0.20	0.21	0.19	0.34	0.37	0.34
Wiki BPE transformer (first sentence, beam=5)	0.39	0.41	0.39	0.22	0.23	0.22	0.37	0.39	0.37
Ria BPE transformer (2000 first tokens)	0.36	0.37	0.35	0.18	0.20	0.18	0.33	0.36	0.33
Textrank summarization	0.14	0.09	0.41	0.05	0.03	0.17	0.09	0.08	0.38
Textrank keywords	0.09	0.07	0.18	0.00	0.00	0.01	0.05	0.05	0.14

Table 2: ROUGE-1,2,F1, precision and recall scores.

After our discovery of a leak in our test set distribution, we decided to check performance of our best abstractive summarization model, transformer with Wikipedia BPE with beam search equal to 5 on dataset with train set leak removed.

Algorithm\Score	1F	1P	1R	2F	2P	2R	LF	LP	LR
Wiki BPE transformer (first sentence, beam=5 with leak)	0.39	0.41	0.39	0.22	0.23	0.22	0.37	0.39	0.37
Wiki BPE transformer (first sentence, beam=5 no leak)	0.39	0.40	0.39	0.22	0.23	0.22	0.36	0.38	0.37

Table 3: Model evaluation without leak

As expected, model performed slightly worse, but still pretty good, so it's still hard to tell, why there is such a great difference between results on out test set and evaluation results from competition servers.

Score	Example
0.00	**Text**: 8 декабря 1991 года россия, белоруссия и украина подписали соглашение о создании содружества независимых государств (снг). **Ground truth**: встреча в беловежской пуще **Prediction**: заявил
0.00	**Text**: более 70 международных художников и арт-групп примут участие в основном проекте "больше света" 5-й московской биеннале современного искусства,который будет показан в цвз "манеж" с 20 сентября по 20 октября, сообщили риа новости в пресс-службе проекта. **Ground truth**: стали известны все участники основного проекта 5-й московской биеннале **Prediction**: более 00 художников представят проект "больше света" в "манеже"
0.99	**Text**: фонд "сколково" и корпорация intel подписали в четверг соглашение о сотрудничестве, передает корреспондент риа новости. **Ground truth**: "сколково" и intel подписали соглашение о сотрудничестве **Prediction**: "сколково" и intel подписали соглашение о сотрудничестве

Table 4: Worse and best model performances. Score is mean of ROUGE-1,2,L F1 score.

6 Related work

The main article, that we've used as an inspiration, was the article VKontakte [2]. From there, we got the idea to use BPE [9], as well as which hyperparameters to use for training of transformer. Then, we've started to browse their citations and we've found that there were a some criticism about what to use as an input to abstractive headline generation algorithms. As a result, we stumbled upon an article where the choice of the first sentence as an input to models was being criticized and authors discussed ways to extract so-called Topic Sentence [7] is proposed.

7 Conclusion and future work

We have seen that the first sentence is a very strong baseline in the problem of generating headers. We have tried various options for preprocessing data, but none of them helped us improve the results of transformer on public leaderboard.

We think that in future work it is worth trying to generate a topic sentence on the 5W1H principle described in [7] or use as an input textrank summarization due to it's high recall.

References

[1] Federico Barrios, Federico López, Luis Argerich, and Rosa Wachenchauzer. Variations of the similarity function of textrank for automated summarization. *CoRR*, abs/1602.03606, 2016.

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

[3] Benjamin Heinzerling and Michael Strube. BPEmb: Tokenization-free Pre-trained Subword Embeddings in 275 Languages. In Nicoletta Calzolari (Conference chair), Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Koiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga, editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 7-12, 2018 2018. European Language Resources Association (ELRA).

[4] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush. OpenNMT: Open-Source Toolkit for Neural Machine Translation. *ArXiv e-prints*.

[5] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Proc. ACL workshop on Text Summarization Branches Out*, page 10, 2004.

[6] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[7] Jan Wira Gotama Putra, Hayato Kobayashi, and Nobuyuki Shimizu. Incorporating topic sentence on neural news headline generation. 2018.

[8] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. http://is.muni.cz/publication/884893/en.

[9] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *CoRR*, abs/1508.07909, 2015.

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.