TED talks summarization

Nikolaevskaya Elena

April 2022

Abstract

This document compares different approaches to summarizing a monologue over a TED movie dataset. Approaches considered: automatic unsupervised text summarization - Text Rank [Kazemi et al., 2020], extractive text summarization - SummaRunner [Nallapati et al., 2016], abstract text summarization - BART [Lewis et al., 2019] Hugging Face. Link to my project code: `https://github.com/dumperize/nlp-project`.

1 Introduction

Nowadays a lot of person face huge flow of information. Daily routine takes away much time, one need to listen to the call recording or meet recording, read articles, listen to podcasts. All of this requires not a little effort and time. I would like to have the opportunity to get information in a compressed form,, and person can save time and energy. In this article will be described a model which can do it.

Firstly, I studied approaches to dialog summarization, I studied works such as [Yuan and Yu, 2019]. But so far my knowledge was not enough to understand the big picture, so I took the task a little easier - instead of a dialogue, I decided to summarize a monologue.That's why I collected the TED dataset [TED, 1984].

1.1 Team

One member - **Nikolaevskaya Elena**.

2 Related Work

Models are divided into supervised and unsupervised methods. Supervised models require summary. It allow the model to learn the scope of the dataset. The summary is usually done by a person, which means that this procedure is expensive.

Another way to classify summarization approaches is extractive and abstractive. In the extractive approach, the summary is assembled from the sentences of the primary document. In the case of the abstractive approach, the model can generate words that are not found in the source text.

The extractive approach is easier to develop, but it is quite limited and it is possible to get unrelated sentences.

An abstractive approach can potentially create good text, but it is much more difficult to develop, and you can get incoherent text even at the word level.

2.1 Automatic unsupervised text summarization

This group does not need anything other than the original article. These methods appeared first and most of them are extractive.

1. Algorithm proposed by Luhn [Luhn, 1958]. The most important words in the text are searched, after which the importance of sentences is calculated based on the presence of important words. The summary includes all sentences whose importance is above the threshold value.

2. TextRank [Kazemi et al., 2020]: Unsupervised Graph-Based Content Extraction. The method was considered in this work.

3. LexRank [Erkan and Radev, 2004] Modified TextRank using similarity measure based on TF-IDF

4. Maximal Marginal Relevance (MMR) [Carbonell and Stewart, 1999] MMR tries to reduce the redundancy of results while at the same time maintaining query relevance of results for already ranked documents/phrases etc.

2.2 Extractive summarization

Extractive summarization selects a subset of sentences from the text to form a summary

1. HAHSum - Neural Extractive Summarization with Hierarchical Attentive Heterogeneous Graph Network [Jia et al., 2020] HAHSum which well models different levels of information, including words and sentences, and spotlights redundancy dependencies between sentences. This approach iteratively refines the sentence representations with redundancy-aware graph and delivers the label dependencies by message passing.

2. BertSumExt [Liu and Lapata, 2019] extraction summarization method based on BERT and reduction to a binary classification problem by the same "oracle" method. Modifications compared to BERT: [CLS] tokens at the beginning of each sentence, interleaved segment embeddingsб MMR-like filtering by 3-gramm, additional Transformer encoder over sentence representations with its own positional embeddings.

3. SummaRuNNer [Nallapati et al., 2016] is one of the first neural network extractive methods for automatic supervised summarization. The method was considered in this work.

2.3 Abstractive summarization

Unlike extractive models, these models create new texts. They can change the original text: delete words or replace them with synonyms, merge and simplify sentences.

1. GPT [Brown et al., 2020] is a set of models (GPT, GPT-2, GPT-3), it are based on the pretraining of the Transformer-decoder. Task is predict the next token from the previous ones, that is, to predict the text from left to right.

2. BertSumAbs [Liu and Lapata, 2019] This model uses the same encoding as BertSumExt. The decoder is a randomly initialized Transformer with 6 layers. As the encoder and decoder have a different number of layers; they have to be train by different optimizers.

   In addition, the authors propose a hybrid BertSumExtAbs scheme: first, we train the encoder on BertSumExt, and then we use it in BertSumAbs.

3. BART [Lewis et al., 2019] - sequence-to-sequence Transformer. The model was considered in this work.

4. PEGASUS [Zhang et al., 2019] - sequence-to-sequence Transformer. But instead of restoring random pieces of text, it is proposed to use the task of generating missing sentences. We select the most important sentences from the document, replace them with a token, form a quasi-abstract from them, and try to generate this quasi-abstract. The authors suggest 3 main strategies to select important sentences: randomly; take the first few sentences; take several sentences of some measure.

3 Models Description

3.1 Automatic unsupervised text summarization - TextRank

TextRank is an automatic referencing method. It is based on the representation of text in the form of an undirected graph. The TextRank algorithm has been improved over the PageRank algorithm. The difference is that the PageRank algorithm creates a network based on the links between web pages, while the TextRank algorithm creates a network according to the relation to word exchange. The main idea is to follow steps:

1. split text into sentences

2. calculate the similarity sentences to each other (these will be edges)

3. build graph sentences with weighted edges

4. calculate the importance of the sentences using the PageRank algorithm, create a summary

Similarity is calculated using the formula below, where S_i is the set of words in the i sentence and S_j is the set of words in the j sentence. It is symmetrical and is calculated in linear time from the number of words.

$$sim_{ij} = \frac{|\{w|(w \in S_i) \wedge (w \in S_j)\}|}{log(|S_i|) + log(|S_j|}$$

The PageRank of a vertex is determined by the formula below, where $P(S_i)$ is the PageRank of the i sentence, sim_{ij} is the similarity between S_i and S_j, S is the set of all sentences, d is the damping factor (0.85 by default). We believe that similar S_j affect S_i more than distant ones.

$$P(S_i) = \frac{(1-d)}{|S|} + d \cdot \sum_{S_j \in S} \frac{sim_{ij}}{\sum_{S_k in S} sim_{ij}} \cdot P(S_j)$$

3.2 Extractive summarization - SummaRunner

The main idea is the “oracle” method, simplifying it to the task of binary classification of sentences. It is necessary to get sentences from the source document so that their summary is as similar as possible to the original abstract for some metrics (for example, for ROUGE). We can make this set greedily: at the very beginning, we choose the first sentence the most likely on original abstract according to our metric, then we choose the second sentence so that the resulting two sentences optimize the metric, and so on. We stop when adding a new offer does not improve the target metric.

The architecture of the model is a two-level bidirectional recurrent neural network, the first level of which passes through words, and the second level through sentences. The scheme is shown in the figure below.

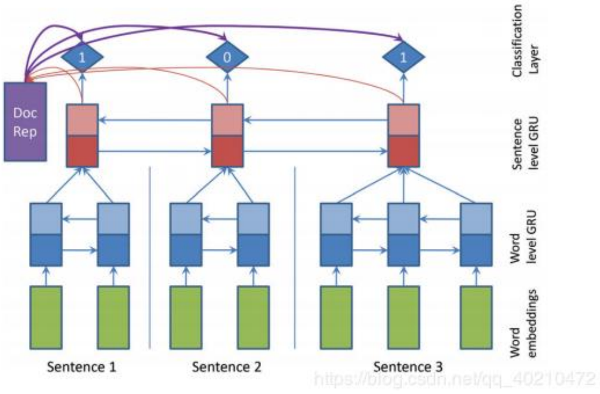

Figure 1: Model SummaRunner.

The model collects document embeddings and uses them in predictions along with network outputs and positional embeddings. Also penalty used in this model.

As a result, the model receives a probability estimate of being included in the final summary for each sentence of the original text. We just need to sort the sentences according to this probability and take the first N as a summary.

The model performed better than all other models in (2016. Important advantage is high speed compared to abstract methods.

3.3 Abstractive summarization - BART

BART is a denoising autoencoder for pretraining sequence-to-sequence models. BART is trained by corrupting text with an arbitrary noising function, and learning a model to reconstruct the original text.

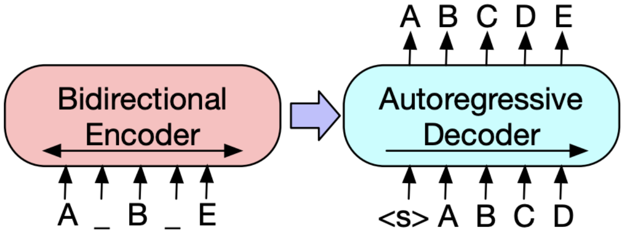

Figure 2: Bart.

It uses a standard Tranformer-based 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.

There are noising approaches, finding the best performance:

- Token Masking Following BERT, random tokens are sampled and replaced with[MASK] elements.
- Token Deletion Random tokens are deleted from the input. In contrast to token masking, the model must decide which positions are missing inputs.
- Text Infilling A number of text spans are sampled, with span lengths drawn from a Poisson distribution.
- Sentence Permutation A document is divided into sentences based on full stops, and these sentences are shuffled in a random order.
- Document Rotation A token is chosen uniformly at random, and the document is rotated so that it begins with that token. This task trains the model to identify the start of the document.

It is important that, unlike BERT, this model is pretrained for text generation, and therefore it is better fit for automatic referencing. Text infilling and document rotation in prelearning problems also helps re-learning.

4 Dataset

Initially, I wanted to investigate the dialog summation task, but it turned out to be quite complicated and for a start I decided that it would be better to start with an easier task - the conversational monologue summation problem. Therefore, the TED resource was chosen as the dataset. There are speeches of people at conferences. The site contains the titles of speeches, short descriptions and transcripts in various languages. This information seemed to me suitable for the purposes of studying the summation task. So I collected data from this site.

5538 video materials were found on the page https://www.ted.com/talks as of April 28, 2022. Previously, I cleaned the dataset, removed the videos that contained:

- empty transcripts
- empty descriptions
- short transcripts (less than 30)
- short descriptions (less than 30)

A total of 3726 materials remained. The resulting dataset is published on my github page.

4.1 EDA

The dataset was examined and the result is shown in the table.

	Text	Summary
Vocabulary size	76206	22953
Lemma Vocabulary size	68124	20754
Common size	17297	

Table 1: Statistics of the TED Dataset. Intersection of two sets of lemmas - summary and text.

As shown in table, the intersection of the lemmas set for the dictionary and the lemmas set for the summary is not covered by the full volume (row “common”). Therefore, abstract approaches will be more suitable for such a dataset.

In this dataset summary length varies a lot (Figure 1). This imposes certain consequences on the calculated metrics. It would be better if it was the same. Might be worth looking into training methods with inconsisten datasets.

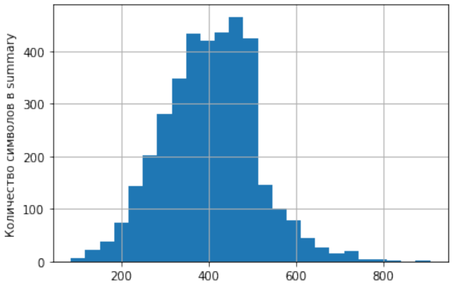

Figure 3: Length summary distribution.

5 Experiments

5.1 Metrics

The most popular metrics for automatic evaluation of Summaries are the ROUGE [Lin, 2004] and BLEU [Callison-Burch et al., 2006] metrics. There are both in this work. The ROUGE metric was calculated in 3 variants: ROUGE-1, ROUGE-2, ROUGE-L.

ROUGE-N is an n-gram ratio between a candidate summary and a set of reference summaries.

ROUGE-L - measures longest matching sequence of words using LCS. An advantage of using LCS is that it does not require consecutive matches but in-sequence matches that reflect sentence level word order. Since it automatically includes longest in-sequence common n-grams, you don't need a predefined n-gram length.

Each of this variants can be calculated for call, precision and F-1. For example, there is the formula for ROUGE-N below.

$$ROUGE-N_{recall} = \frac{Num\ words\ matches}{Num\ words\ in\ reference}$$

$$ROUGE-N_{precision} = \frac{Num\ words\ matches}{Num\ words\ in\ summary}$$

$$ROUGE-N_{F1} = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

The reason one would use ROUGE-1 over or in conjunction with ROUGE-2 (or other), is to also show the fluency of the summaries.

BLEU is metric for evaluating a generated sentence to a reference sentence like a $ROUGE-N_{precision}$. BLEU is to compare different n-grams of the candidate with the different n-grams of the reference together and count the number of matches. Because BLEU is precision based, a brevity penalty is introduced to compensate for the possibility of proposing highprecision hypothesis which are too short.

The penalty is calculated as:

$$BP = \begin{cases} 1, & \text{if } c > r \\ e^{1-r/c}, & \text{if } c \leq r \end{cases}$$

where c is the length of the corpus of hypothesis translations, and r is the effective reference corpus length.

BLEU score is calculated as:

$$BLEU = BP \cdot \exp \sum_{n=1}^{N} w_n \log \frac{\sum_n num\, ngram\, matches}{\sum_n num\, ngram\, in\, reference}$$

The difference between the ROUGE-n precision and BLEU is that BLEU introduces a brevity penalty term, and also compute the n-gram match for several size of n-grams (unlike the ROUGE-n, where there is only one chosen n-gram size).

5.2 Experiment Setup

In total, 6 approaches to solving the problem were carried out: LEAD-3, TextRank, library Summa, SummaRunner, Bart without trained, Bart with trained.

Self-written solutions were used for Lead-3, TextRank, Summarunner. For the BART model, the implementation from the Hugging Face preterminated model based on news (bart-large-cnn) was used. The blurr library was used to train the BART model.

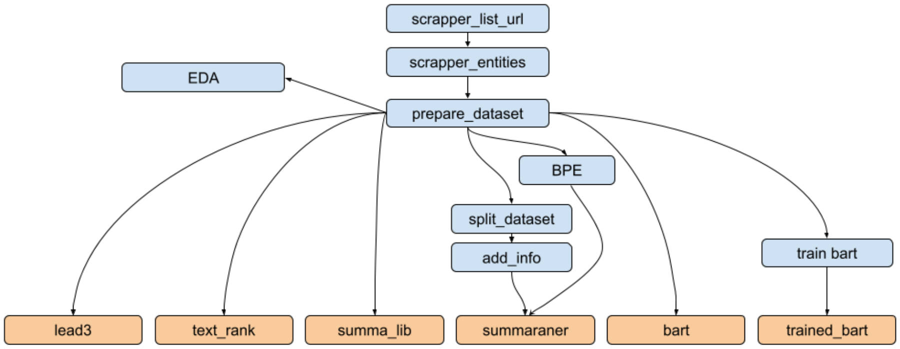

Figure 4: The job DAG

5.3 Baselines

Traditionally, baseline for text summarization is the Lead-3 model. In the model the first 3 sentences is predicted summary. This approach gives not bad results for news dataset. And on my dataset, this approach is quite good too. The results are presented in the chapter Results

6 Results

The table below shows the metrics for different approaches. As expected for this dataset, the best quality was obtained for abstractive approach and the trained BART model.

		Lead 3	TextRank	SummaRunner	BART	Trained BART
BLEU		0.30	0.24	0.33	0.37	0.43
ROUGE-1	f	0.23	0.19	0.26	0.23	0.29
	p	0.28	0.17	0.31	0.21	0.28
	r	0.20	0.31	0.24	0.28	0.31
ROUGE-2	f	0.08	0.03	0.08	0.05	0.10
	p	0.10	0.03	0.10	0.05	0.09
	r	0.07	0.05	0.07	0.06	0.11
ROUGE-L	f	0.20	0.17	0.22	0.19	0.25
	p	0.25	0.16	0.27	0.18	0.24
	r	0.18	0.22	0.20	0.22	0.27

Table 2: Comparing different approach.

In the EDA section, it was shown that in the dataset, the human-created summary varies greatly in length. This fact contributes to the metrics. Perhaps it was worth trying to divide the dataset into pieces with short average and long sums and run the models separately.

It is also worth noting that in almost all summaries there are phrases such as "Person X told in his speech about ...." To improve the quality, it might be worth pointing out the author's abstractive summarization models and teaching similar speech constructions.

7 Conclusion

In this work, various approaches to summarizing texts were considered, such as automatic unsupervised text summarization, extractive summarization and abstractive summarization.

For experiments, a dataset was collected from the TED website. The evaluation was carried out according to the metrics of ROUGE and BLEU, the best quality was obtained on the trained model of the abstractive approach of BART.

References

[Brown et al., 2020] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. arXiv.

[Callison-Burch et al., 2006] Callison-Burch, C., Osborne, M., and Koehn, P. (2006). Re-evaluating the role of Bleu in machine translation research. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 249–256, Trento, Italy. Association for Computational Linguistics.

[Carbonell and Stewart, 1999] Carbonell, J. and Stewart, J. (1999). The use of mmr, diversity-based reranking for reordering documents and producing summaries. *SIGIR Forum (ACM Special Interest Group on Information Retrieval)*.

[Erkan and Radev, 2004] Erkan, G. and Radev, D. R. (2004). LexRank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22:457–479.

[Jia et al., 2020] Jia, R., Cao, Y., Tang, H., Fang, F., Cao, C., and Wang, S. (2020). Neural extractive summarization with hierarchical attentive heterogeneous graph network. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3622–3631, Online. Association for Computational Linguistics.

[Kazemi et al., 2020] Kazemi, A., Pérez-Rosas, V., and Mihalcea, R. (2020). Biased textrank: Unsupervised graph-based content extraction. arXiv.

[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. arXiv.

[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.

[Liu and Lapata, 2019] Liu, Y. and Lapata, M. (2019). Text summarization with pretrained encoders. arXiv.

[Luhn, 1958] Luhn, H. P. (1958). The automatic creation of literature abstracts. volume 2, pages 159–165.

[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.

[TED, 1984] TED (1984). Ted conferences, llc (technology, entertainment, design).

[Yuan and Yu, 2019] Yuan, L. and Yu, Z. (2019). Abstractive dialog summarization with semantic scaffolds. arXiv.

[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. arXiv.