Topical extractive summarization

Olga Medvedeva

January 2023

Abstract

Thematic extractive generalization is aimed at highlighting the sentences that are most relevant to a given topic. There are many approaches to creating a resume from source text. However, only a few works are devoted to extractive generalization using any information on topics. This article proposes a vector cosine similarity approach combined with topic modeling to generate a summary of a topic. Link to project code: `https://github.com/olya-ds/topical-extractive-summarization`.

1 Introduction

The task of summarizing text has become popular in NLP because it affects many areas of our lives. When working and communicating on the Internet, the user wants to read a short version of the text before reading it in its entirety to save time and effort. Summarization is divided into two types: extractive and abstract. Extractive summarization compiles a summary of the text from the sentences found in the original text. An abstract (or generative) summarization of a text consists of a mixture of sentences with all the important details conveyed. Thematic extractive summarization is only part of the extractive summarization for the compilation of short texts from the sentences most relevant to the topic. A topic is a list of keywords that are most likely to be related to a topic. To do this, Latent Dirichlet Allocation (LDA, Blei and Lafferty (2009)) is used, which divides texts into clusters in such a way that each cluster contains texts with similar topics.

Recently, neural networks, transformers, as well as the TextRank method and methods based on it have been used to summarize the text. In this paper, after topic modeling and obtaining topic keywords, the cosine similarity of text sentence vectors and topic keyword vectors is calculated to find the sentences most similar to the topic for the summary.

2 Related Work

There are few works on the subject of thematic extractive summarization in the sources. But there are many works separately about extractive summarization.

One of the first such works is [Mihalcea and Tarau, 2004]. In this paper describe TextRank, a graph ranking model for text processing and sentence extraction. The vertices of the graph are the sentences, and the links between them are the similarity of the sentences. The basic idea is that when one vertex binds to another, it essentially votes for that other vertex. The higher the number of votes cast for a top, the higher the importance of the top.

In [Gunes Erkan, 2011], the authors propose a graph-based sentence ranking algorithm for extractive summation. This method is a version of the TextRank algorithm extended for the DUC 2006 focused summation problem. As in TextRank, the set of sentences in a cluster of documents is represented as a graph, where nodes are sentences, and links between nodes are induced similarity relationships between sentences. The authors use a similarity measure based on TF-IDF and vanish graph edges with a weight below a certain threshold.

In the work [R. Nallapati, 2016] authors present Recurrent Neural Network based on sequence model for extractive summarization. The work focuses only on sentential extractive summarization of single documents using neural networks. Extractive summarization is treated as a sequence classification problem wherein, each sentence is visited sequentially in the original document order and a binary decision is made in terms of whether or not it should be included in the summary. GRU based RNN was used as the basic building block of the sequence classifier.

In the [J. Xu, 2020] work the discourse-aware neural extractive summarization model was built upon BERT. To perform compression with extraction simultaneously and reduce redundancy across sentences, authors take Elementary Discourse Unit (EDU) as the minimal selection unit (instead of sentence) for extractive summarization. Extractive summarization is formulated as a sequential labeling task, where each EDU is scored by neural networks, and decisions are made based on the scores of all EDUs.

The authors of the paper [S. Ullah, 2019] propose a novel approach for generating extractive summary by considering semantic relationships among the sentences of the document. He extract the Predicate Argument Structure (PAS) from each sentence and measure semantic distance among the sentences is measured from the PAS to PAS semantic relationship of the sentences. Then, we apply modified page rank algorithm to rank the sentences. Subsequently, we employ Maximum Marginal Relevance (MMR) method to rerank the sentences.

In the article [D. Parveen, 2015], the authors present an approach to extractive generalization of individual documents. Their approach is based on a weighted graphical representation of documents obtained through topic modeling. We optimize importance, consistency, and lack of redundancy at the same time with ILP.

3 Model Description

The algorithm is to tokenize the input articles, remove stop words, get bigrams, trigrams and lemmas of words. Then compile a dictionary and corpus for Latent

Dirichlet placement. At the output of the model, we get the keywords of each topic. The algorithm described above is schematically presented in Fig. 1.

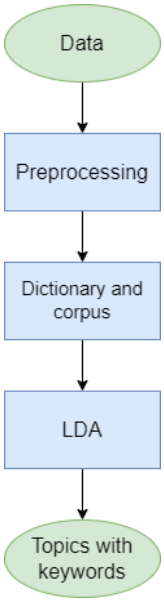

Figure 1: Topic modeling scheme.

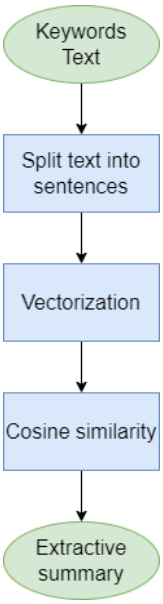

Figure 2: Summarization scheme.

After that, you can calculate the main topic in the document and the probability of referring the document to each of the list of topics. At the next stage, each document is divided into sentences and vectorized. Between the vectorized sentences and the vectorized keywords of the main topic of the current document, cosine similarity is calculated:

$$S = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}},$$

where S is a cosine similarity, A_i and B_i are components of vector A and B respectively.

The top 3 sentences of a document by similarity to the theme's keyword vector constitute the document's predictive summary (Fig. 2). The order of the sentences in the summarized text corresponds to the order in the original text.

4 Dataset

The dataset used is the BBC News Summary[1], which consists of BBC news articles and extractive summaries of the news. The dataset includes 2225 articles. They are divided into 5 topics: business, entertainment, sports, politics, tech. This dataset was chosen because it is in the public domain and the articles in it are divided into topics and it is interesting to see what the prevailing topics are according to the topic model.

Instead of the original markup of the dataset, the markup was prepared independently using a pre-trained transformer[2]. The input is articles from the dataset, and the output is a summary of these articles. At the same time, output summaries of articles are recorded in the csv file.

5 Experiments

5.1 Metrics

5.1.1 Topic modeling metrics

Topic **coherence** provide a convenient measure to judge how good a given topic model is. The coherence score in topic modeling measures how people interpret topics. In this case, topics are presented as the first N words most likely to belong to that particular topic. The coherence score measures how similar these words are to each other. One of the most popular indicators of coherence is called cv. It creates word content vectors using their matches and then computes a score using normalized pointwise mutual information (NPMI) and cosine similarity.

5.1.2 Extractive summarization metrics

The metric for evaluating the quality of text summarization is Recall-Oriented Understudy for Gisting Evaluation (ROUGE). ROUGE is a set of metrics. In this work, ROUGE-N and ROUGE-L are used. ROUGE-N measures the number of matching 'n-grams' between predicted summary and a 'reference' it consists of recall, precision and F1-score. An n-gram is simply a grouping of tokens/words. ROUGE-1 and ROUGE-2 measures the degree of unigram and bigrams match between the predicted summary and the reference respectively.

The **recall** counts the number of overlapping n-grams found in both the predicted summary and reference — then divides this number by the total number of n-grams in the reference. It looks like this:

$$Recall = \frac{count_{match}(gram_n)}{count(gram_n)},$$

[1] https://www.kaggle.com/datasets/pariza/bbc-news-summary.
[2] https://huggingface.co/sshleifer/distilbart-cnn-12-6.

The disadvantage of the recall is that it does not take into account the number of words in the prediction. This can be solved with a **precision** metric that looks almost the same, but divided by the number of words in the prediction.

Now that we both the recall and precision values, we can use them to calculate our ROUGE-N **F1-score** like so:

$$F1 = 2 * \frac{precision * recall}{precision + recall},$$

That gives us a reliable measure of our model performance that relies not only on the model capturing as many words as possible (recall) but doing so without outputting irrelevant words (precision).

ROUGE-L measures the longest common subsequence (LCS) between the output summary and the reference. All this means is that the longest sequence of tokens that is used by both is considered. The idea here is that a longer overall sequence will indicate more similarity between the two sequences. You can apply recall and precision calculations as before, but this time change the match to LCS.

5.2 Experiment Setup

There were 6 launches with a different number of topics. The topic model is trained with such parameters:

- number of topics - [2, 5, 8, 11, 14, 17]
- number of documents to be used in each training chunk - 100
- number of passes through the corpus during training - 10
- number of documents to be iterated through for each update -1
- computes a list of topics, sorted in descending order of most likely topics for each word, along word count - True

Each topic is represented as a list of 10 top tokens. The results of the experiments are presented in the Tab. 1.

Num topics	Coherence
2	0.466
5	0.503
8	0.484
11	0.447
14	0.430
17	0.422

Table 1: Statistics of experiments on the number of topics.

6 Results

The optimal number of topics corresponds to the number of topics in the dataset. The top tokens of the best model are used to calculate the cosine similarity between the keywords of the probable article topic and the article sentence. Between each predicted summary and the actual summary, ROUGE-1, ROUGE-2 and ROUGE-L marks are calculated. Then the average of the metrics for all texts is calculated. The result is presented in Tab. 2. For the BBC News

Metrics	Rouge-1	Rouge-2	Rouge-L
F1	50.8	35.3	49.0
Recall	49.2	33.9	47.5
Precision	53.1	37.2	51.2

Table 2: Average metrics for all texts.

dataset, there are no articles with a solution to the problem of topical extraction summarization. There are [S. Ullah, 2019] with the solution of the problem of extractive summarization on the BBC News dataset. Comparisons of the results of ROUGE-2 of this article with other methods are transferred to Tab. 3. In the

Method	Recall	Precision	F-measure
Proposed	31	31	31
LexRank	28	25	26.41
LSA	29	27.23	28
TextRank	26	23.78	24.84

Table 3: Recall, precision and f-measure for Rouge-2 in [S. Ullah, 2019].

paper [E. Zolotareva and Horvath, 2020] dataset is used to train models (T5, Seq2Seq) to solve the problem of abstractive summarization. The results of this article are presented in Tab. 4 and Tab. 5. When solving this problem on the

Metrics	Rouge-1	Rouge-2	Rouge-L
F1	47.3	26.5	36.1
Recall	48.0	26.9	38.9
Precision	46.7	26.1	33.8

Table 4: Results on the BBC test set using T5 Model in [E. Zolotareva and Horvath, 2020].

DUC 2002 dataset, the results were achieved in accordance with Tab. 6.

In Tab. 7 examples of the work of the proposed algorithm are shown. Predicted and actual summaries of news articles.

Metrics	Rouge-1	Rouge-2	Rouge-L
F1	32.3	19.3	26.2
Recall	32.4	13.2	19.9
Precision	38.8	27.5	28.9

Table 5: Results on the BBC test set using Seq2Seq Model in [E. Zolotareva and Horvath, 2020].

Systems	Rouge-1	Rouge-2
Lead	45.9	18.0
DUC 2002 Best	48.0	22.8
TextRank	47.0	19.5
UniformLink (k = 10)	47.1	20.1
Egraph + Coh.	47.9	23.8
Tgraph (n=2000) + Coh.	48.1	24.3

Table 6: Results on the BBC test set using Seq2Seq Model in [D. Parveen, 2015].

Japan's economy teetered on the brink of a technical recession in the three months to September, figures show.Revised figures indicated growth of just 0.1 - and a similar-sized contraction in the previous quarter.On an annual basis, the data suggests annual growth of just 0.2, suggesting a much more hesitant recovery than had previously been thought.
Japan's economy teetered on the brink of a technical recession in the three months to September. Growth of just 0.1 - and a similar-sized contraction in the previous quarter. Annual growth of 0.2 suggests a much more hesitant recovery than previously thought. The government plays down the worrying implications of the data, saying it is in a 'minor adjustment phase'.

Table 7: Output example of predicted and actual news article summary.

7 Conclusion

Summing up the work, markup was made for the BBC News dataset, topic modeling was done, and the resulting keywords were used to measure cosine similarity with document sentences. This algorithm made it possible to achieve averages over the entire dataset of metrics more than other methods and models.

References

[D. Parveen, 2015] D. Parveen, H. Ramsl, M. S. (2015). Topical coherence for graph-based extractive summarization. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, page 1949–1954.

[E. Zolotareva and Horvath, 2020] E. Zolotareva, T. M. T. and Horvath, T. (2020). Abstractive text summarization using transfer learning.

[Gunes Erkan, 2011] Gunes Erkan, D. R. R. (2011). Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal Of Artificial Intelligence Research*, 22:457–479.

[J. Xu, 2020] J. Xu, Z. Gan, Y. C. J. L. (2020). Discourse-aware neural extractive text summarization.

[Mihalcea and Tarau, 2004] Mihalcea, R. and Tarau, P. (2004). Textrank: Bringing order into text. *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411.

[R. Nallapati, 2016] R. Nallapati, F. Zhai, B. Z. (2016). Summarunner: A recurrent neural network based sequence model for extractive summarization of documents.

[S. Ullah, 2019] S. Ullah, A. I. (2019). A framework for extractive text summarization using semantic graph based approach. *6th International Conference on Networking, Systems and Security*.