CNLP-NITS @ LongSumm 2021: TextRank Variant for Generating Long Summaries

Darsh Kaushik, Abdullah Faiz Ur Rahman Khilji, Utkarsh Sinha, Partha Pakray
Department of Computer Science and Engineering
National Institute of Technology Silchar
Assam, India
{darsh_ug, abdullah_ug, utkarsh_ug, partha}@cse.nits.ac.in

Abstract

The huge influx of published papers in the field of machine learning makes the task of summarization of scholarly documents vital, not just to eliminate the redundancy but also to provide a complete and satisfying crux of the content. We participated in LongSumm 2021: The 2^{nd} Shared Task on Generating Long Summaries for scientific documents, where the task is to generate long summaries for scientific papers provided by the organizers. This paper discusses our extractive summarization approach to solve the task. We used TextRank algorithm with the BM25 score as a similarity function. Even after being a graph-based ranking algorithm that does not require any learning, TextRank produced pretty decent results with minimal compute power and time. We attained 3^{rd} rank according to ROUGE-1 scores (0.5131 for F-measure and 0.5271 for recall) and performed decently as shown by the ROUGE-2 scores.

1 Introduction

Text summarization or summarizing large pieces of texts into comparatively smaller number of words is a challenging machine learning (ML) task that has gained significant traction in recent years. The applications are immense and diverse, from condensing and comparing legal contractual documents to summarizing medical and clinical texts. Often the two approaches (Maybury, 1999) adopted for solving this task are:

- Extractive summarization:
  Here those unmodified segments of the original text are extracted and concatenated which play the most significant role in expressing the salient sentiment of the entire text. This technique is mostly used for generating comparatively longer summaries.
- Abstractive summarization:
  Here an abstract semantic representation of the original content is formed by the model which helps generate novel words/phrases for the summary by text generation and paraphrasing methods. This technique is often useful for generating concise summaries.

Recently, the task of summarizing scholarly documents has grasped the attention of researchers due to the vast quantity of papers published everyday, especially in the field of machine learning. This makes it challenging for researchers and professionals to keep-up with the latest developments in the field. Thus, the task of summarizing scientific papers aims not just to avoid redundancy in text and generate shorter summaries but also to cover all the salient information present in the document which often demands longer summaries. This would aid researchers to grasp the contents of the paper beyond abstract-level information without reading the entire paper.

Prior work on summarization of scientific documents is mostly targeted towards generation of short summaries but as mentioned before, in order to encompass all the important ideas longer summaries are required. LongSumm 2021[1] shared task, on the other hand, aims to encourage the researchers to focus on generating longer-form summaries for scientific papers.

As mentioned before, extractive summarization methods are better accustomed for generating longer-form summaries than abstractive summarization methods, in this paper we try to summarize scientific documents using the extractive summarization technique of TextRank (Mihalcea and Tarau, 2004) algorithm. It is a graph-based ranking algorithm to rank the sentences in a document according to their importance in conveying the

[1] https://sdproc.org/2021/sharedtasks.html

Proceedings of the Second Workshop on Scholarly Document Processing, pages 103–109

information of the document. Different 'similarity' functions can be used while creating the graph which leads to varied results (Barrios et al., 2016), therefore we chose BM25 as the similarity function.

2 Related Works

Upon scrutinizing various approaches of document summarization, we have found some of the concurrent works in the field. One of these is (Christensen et al., 2013). This work describes extractive summarization as a joint process of selection and ordering. It uses graph as its elemental part, which is used to approximate the discourse relativeness using co-reference, deverbal nouns, etc. Similar works are shown by (Li et al., 2011), (Goldstein et al., 2000) and (Barzilay et al., 1999). Other works use the Google TextRank algorithm (Mihalcea and Tarau, 2004) to bring out the order in the text extraction. One of the works (Mallick et al., 2019) uses the modified TextRank plus graph infrastructure to extract contextual information. It uses the sentence as nodes in the graph and inverse cosine similarity[2] to form the weights of the edges of the graph. This graph is passed as an input to the TextRank algorithm which generates the required summary. Similar approach is followed by (Ashari and Riasetiawan, 2017) which uses the power of TextRank and semantic networks to form extractive summaries which bear the semantic relations.

Some of the works like (Nallapati et al., 2017), (Al-Sabahi et al., 2018) use capabilities of neural networks to semantically extract the information from the description and present it in human readable form. One of the works (Nallapati et al., 2016) uses a joint framework of classification and selection on the textural data to form summaries. Classifier architecture makes a decision as to whether a particular sentence in sequence (as selected by selector) will be the part of the membership of the summary or not, whereas the selector framework randomly selects the sentences from the description and places it in the summary.

Apart from these, varied approaches were adopted by the participants of the previous edition of the shared task, LongSumm 2020, as mentioned in (Chandrasekaran et al., 2020). For instance, a divide and conquer approach, DANCER, was used in (Gidiotis et al., 2020) to summarize key sections of the paper separately and combine them through a PEGASUS based transformer to generate the final summary. Another team (Ghosh Roy et al., 2020) used a neural extractive summarizer to summarize each section separately. A different team utilized the BERT summarizer as shown in (Sotudeh Gharebagh et al., 2020). The main idea was based on multi-task learning heuristic in which two tasks are optimized, namely the binary classification task of sentence selection and the section prediction of input sentences. They also suggested an abstractive summarizer based on the BART transformer that runs after the extractive summarizer. Other methods were Convolutional Neural Network (CNN) in (Reddy et al., 2020), Graph Convolutional Network (GCN) and Graph Attention Network (GAN) in (Li et al., 2020), and unsupervised clustering in (Mishra et al., 2020) and (Ju et al., 2020).

3 Dataset

3.1 Description

The LongSumm dataset is distinctive in the sense that it consists of scientific documents which have scientific jargon targeted for a niche audience, unlike other summarization corpuses like news articles for the general public. Due to the same reason, it is difficult to find domain-specific scientific documents with their longer-form summaries covering all their important details in a concise manner.

The organizers of LongSumm 2021 provided corpus for this task includes a training set that consists of 1705 extractive summaries, and 531 abstractive summaries of NLP and Machine Learning scientific papers. The extractive summaries are based on video talks (Chandrasekaran et al., 2020) from associated conferences while the abstractive summaries are blog posts created by NLP and ML researchers.

We used Textrank (Mihalcea and Tarau, 2004) which is a graph-based ranking model for ranking sentences in a document for extractive summarization. Therefore, only extractive summaries were used as validation data. The extractive summaries

[2] https://link.springer.com/chapter/10.1007/978-3-642-41278-3_74

are based on the TalkSumm (Lev et al., 2019) dataset. The dataset contains 1,705 automatically generated noisy extractive summaries of scientific papers from the NLP and Machine Learning domain based on video talks from associated conferences (like ACL, NAACL, ICML). URL links to the papers and their summaries and could be found in the Github repository[3] devoted to this shared task. Each summary provides the top-30 sentences, which are on average around 990 words.

Another list of 22papers [4] was provided as test data (blind). The summaries generated for these papers were used for evaluation. ROUGE-1, ROUGE-2 and ROUGE-L scores were used to evaluate the performance of the system.

3.2 Preprocessing

After retrieving the text from the papers (links to which were provided by the organizers) the sections before 'Introduction' (like Authors, Abstract etc.) and after 'Conclusion/Results' (like References, Acknowledgements etc.) were removed as the text in these sections do not add much valuable sentiments to the summary as compared to the left over sections of the paper. Further citation indexing, hyperlinks, newline and redundant white-space characters were eliminated.

4 System Description

Our approach essentially was to use the TextRank algorithm (Mihalcea and Tarau, 2004) to rank the sentences corresponding to their relevance to the whole text and use the most significant (highest ranked) sentences as the summary.

4.1 TextRank

TextRank is a graph-based ranking algorithm which is proven to be quite impactful for keyword and sentence extraction from natural language texts.

According to (Mihalcea and Tarau, 2004) for sentence extraction, a graph is constructed for the given document in which each vertex represents an entire sentence. Now the semantic links amongst the vertices are identified by the "similarity" between the sentences, where "similarity" is measured as a function of their content overlap. The formal expression for determining the similarity of two sentences, $S_a = w_1^a, w_2^a, ..., w_{N_a}^a$ with N_a words, and $S_b = w_1^b, w_2^b, ..., w_{N_b}^b$ with N_b words as defined in (Mihalcea and Tarau, 2004):

$$Sim(S_a, S_b) = \frac{|\{w_k | w_k \in S_a \& w_k \in S_b\}|}{log(|S_a|) + log(|S_b|)}$$

The text in the document can thus be represented as a weighted on which the ranking algorithm is run to sort the vertices (each representing a sentence in the text) in reversed order of the obtained score, from which we include the 30 most significant sentences are selected and present them in the same order as they appear in the document.

4.2 Gensim TextRank Summarizer

Variants of the similarity function can be chosen to obtain improved results, an analysis of which is shown in (Barrios et al., 2016). The different similarity functions including LCS (Longest Common Substring), cosine similarity, BM25 (Robertson et al., 1994), BM25+ (Lv and Zhai, 2011) and original TextRank similarity function were evaluated using ROUGE-1, ROUGE-2 and ROUGE-SU4 as metrics in (Barrios et al., 2016) and the best results were obtained using BM25 and BM25+.

The Summarizer module of the Gensim project [5] uses BM25-TextRank algorithm for summarization, therefore we proceeded with this implementation of TextRank to prepare the summaries. BM25 is a variation of the TF-IDF model using a probabilistic model. Given two sentences R, S, BM25 is defined as:

$$BM25(R,S) = \sum_{i=1}^{n} IDF(s_i) \cdot \frac{TF(s_i, R) \cdot (k+1)}{TF(s_i, R) + k \cdot (1 - b + b \cdot \frac{|R|}{L_{avg}})}$$

where k and b are parameters, and L_{avg} is the average length of the sentences in the document. TF is the term-frequency and IDF is the correction formula given as:

$$IDF(s_i) = \begin{cases} log(\frac{N - n(s_i) + 0.5}{n(s_i) + 0.5}) & \text{if } n(s_i) > N/2 \\ \epsilon \cdot avgIDF & \text{if } n(s_i) \leq N/2 \end{cases}$$

[3] https://github.com/guyfe/LongSumm/tree/master/extractive_summaries

[4] https://github.com/guyfe/LongSummtest-data-blind

[5] https://github.com/summanlp/gensim

where ϵ takes a value between 0.5 and 0.3 and avgIDF is the average IDF for all terms.

5 Result and Analysis

5.1 Result

The participating systems were evaluated by ROUGE(Lin, 2004) scores, specifically using ROUGE-1, ROUGE-2 and ROUGE-L metrics. Our team's name was CNLP-NITS and the result of our system on blind test data of 22 papers using TextRank with BM25 similarity is given in Table 1.

Metric	F-measure	Recall
ROUGE-1	0.5131	0.5271
ROUGE-2	0.161	0.1656
ROUGE-L	0.1916	0.1971

Table 1: ROUGE scores for blind test data

As 22 papers was not a large dataset, we also applied TextRank on the given dataset of extractive summaries (1700 of them) to get statistically sound ROUGE scores for analysis, and the scores obtained are shown Table 2.

Metric	F-measure	Recall
ROUGE-1	0.59389	0.5960
ROUGE-2	0.3349	0.3362
ROUGE-L	0.3393	0.3405

Table 2: ROUGE scores for training dataset of extractive summaries

5.2 Analysis

Individual ROUGE scores for each paper in the training set was calculated for finding the average scores.

The predicted and reference summary for the paper [6] with one of the best ROUGE scores (as given in Table 3) are as shown,

Metric	F-measure	Recall
ROUGE-1	0.88	0.88
ROUGE-2	0.8164	0.8164
ROUGE-L	0.8217	0.8217

Table 3: ROUGE scores for predicted summary of the paper[6] with one of the best performances

Predicted summary (best performance):

"Over the past few years neural models based on the encode-attend-decode (Bahdanau et al., 2014) paradigm have shown great success in various natural language generation (NLG) tasks such as machine translation (Bahdanau et al., 2014), abstractive summarization ((Rush et al., 2015),(Nallapati et al., 2016)) dialog (Li et al., 2016), etc. One such NLG problem which has not received enough attention in the past is query based abstractive text summarization where the aim is to generate the summary of a document in the context of a query. Thus given a document on "the super bowl", the query "How was the half-time show?", would result in a summary that would not cover the actual game itself. Note that there has been some work on query based extractive summarization in the past where the aim is to simply extract the most salient sentence(s) from a document and treat these as a summary. Since, we were interested in abstractive (as opposed to extractive) summarization we created a new dataset based on debatepedia. This dataset contains triplets of the form (query, document, summary)...."[7]

Reference summary (best performance):

"Over the past few years neural models based on the encode-attend-decode (Bahdanau et al., 2014) paradigm have shown great success in various natural language generation (NLG) tasks such as machine translation (Bahdanau et al., 2014), abstractive summarization ((Rush et al., 2015),(Nallapati et al., 2016)) dialog (Li et al., 2016), etc. One such NLG problem which has not received enough attention in the past is query based abstractive text summarization where the aim is to generate the summary of a document in the context of a query. In general, abstractive summarization, aims to cover all the salient points of a document in a compact and coherent fashion. On the other hand, query focused summarization highlights those points that are relevant in the context of the query. Thus given a document on the super bowl, the query How was the half-time show?, would result in a summary that would not cover the actual game itself. Note that there has been some work on query based extractive summarization in the past where the aim is to simply extract the most salient sentence(s) from a document and treat these as a summary...."[8]

The predicted and reference summary for the paper[11] with one of the worst ROUGE scores (as given in Table 4) are as shown,

[6] https://www.aclweb.org/anthology/P17-1098.pdf

[7] Complete summary at https://bit.ly/3914zEy

[8] Complete summary at https://bit.ly/3c5elHQ

Metric	F-measure	Recall
ROUGE-1	0.305	0.305
ROUGE-2	0.0301	0.0301
ROUGE-L	0.1217	0.1217

Table 4: ROUGE scores for predicted summary of the paper[11] with one of the worst performances

Predicted summary (worst performance):

"Although state-of-the-art sensors can detect various natural disasters in advance (e.g., Mexico City's alarm system can timely sense earthquakes originating in the southern states) [1], the devastating consequences of these events in urban areas are usually severe. As an example, in the 2010 earthquake in Haiti, the use of instant messages sent by civilians from different locations facilitated the reporting of trapped individuals, the provision of medical assistance, and the delivery of basic needs, such as food, water, and shelter [5]. Personal mobile devices can be linked to Online Social Networks (OSNs) and enable synchronization among applications, e.g., Twitter, Facebook, and Instagram, which allows users to post and update their activities in real time [7,8]. A tweet providing the location (spatial information) of a collapsed building, along with a timestamp (temporal information), one day after the 2017 earthquake in Mexico City. Recently, Twitter has been the center of attention in different research fields related to Marketing, Social Sciences, Natural Language Processing (NLP), Opinion Mining, and Predictive analysis [17]...."[9]

Reference summary (worst performance):

"Second, dividing a corpus into separate time bins may lead to training sets that are too small to train a word embedding model. Hence, one runs the risk of overfitting to few data whenever the required temporal resolution is fine-grained, as we show in the experimental section. We show the ten words whose meaning changed most drastically in terms of cosine distance over the last 150 years. We thereby automatically discover words such as computer or radio whose meaning changed due to technological advances, but also words like peer and notably whose semantic shift is less obvious. Their approach uses a non-Bayesian treatment of the latent embedding trajectories, which makes the approach less robust to noise when the data per time step is small. trained end-to-end and scales to massive data by means of approximate Bayesian inference. For each pair of words i, j in the vocabulary, the model assigns probabilities that word i appears in the context of word j. ..."[10]

[9]Complete summary at https://bit.ly/3cWCmjo

[10]Complete summary at https://bit.ly/3f2XO9i

6 Conclusion and Future Work

In this paper we targeted our efforts towards TextRank algorithm in order to generate long extractive summaries of given scientific research papers. Our approach TextRank when used with BM25 similarity function, even after not being a learning algorithm, was able to achieve appreciable ROUGE-1 scores while remaining competitive in ROUGE-2 scores. As TextRank is a graph-based ranking algorithm that ranks the sentences independently for each document, it requires no training, thus being compute and time efficient.

Although we approached the task using an algorithm which does not require training and were still able to produce substantial results, there is definitely a scope for leveraging training data to gather a general semantic structure from a collection of documents as a whole instead of working on each document independently using neural network based learning algorithms. This will definitely be our prime focus for future work in extractive text summarization. Nonetheless, through our participation in LongSumm 2021 we tried to optimise TextRank algorithm and put it to test against other learning-based approaches of other teams and were able to pull off significant results with comparatively low machine and time requirements.

Acknowledgements

We would like to thank Department of Computer Science and Engineering and Center for Natural Language Processing (CNLP) at National Institute of Technology Silchar for providing the requisite support and infrastructure to execute this work. The work presented here falls under the Research Project Grant No. IFC/4130/DST-CNRS/2018-19/IT25 (DST-CNRS targeted program). The authors would also like to thank LongSumm 2021 shared task organizers for organizing this event.

References

Kamal Al-Sabahi, Zhang Zuping, and Mohammed Nadher. 2018. A hierarchical structured self-attentive model for extractive document summarization (hssas). *IEEE Access*, 6:24205–24212.

Ahmad Ashari and Mardhani Riasetiawan. 2017. Document summarization using textrank and semantic network. *International Journal Intelligent Systems and Applications*, pages 26–33.

[11]https://www.mdpi.com/1424-8220/19/7/1746

Federico Barrios, Federico López, Luis Argerich, and Rosa Wachenchauzer. 2016. Variations of the similarity function of textrank for automated summarization.

Regina Barzilay, Kathleen McKeown, and Michael Elhadad. 1999. Information fusion in the context of multi-document summarization. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics*, pages 550–557.

Muthu Kumar Chandrasekaran, Guy Feigenblat, Eduard Hovy, Abhilasha Ravichander, Michal Shmueli-Scheuer, and Anita de Waard. 2020. Overview and insights from the shared tasks at scholarly document processing 2020: CL-SciSumm, LaySumm and LongSumm. In *Proceedings of the First Workshop on Scholarly Document Processing*, pages 214–224, Online. Association for Computational Linguistics.

Janara Christensen, Stephen Soderland, Oren Etzioni, et al. 2013. Towards coherent multi-document summarization. In *Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: Human language technologies*, pages 1163–1173.

Sayar Ghosh Roy, Nikhil Pinnaparaju, Risubh Jain, Manish Gupta, and Vasudeva Varma. 2020. Summaformers @ LaySumm 20, LongSumm 20. In *Proceedings of the First Workshop on Scholarly Document Processing*, pages 336–343, Online. Association for Computational Linguistics.

Alexios Gidiotis, Stefanos Stefanidis, and Grigorios Tsoumakas. 2020. AUTH @ CLSciSumm 20, LaySumm 20, LongSumm 20. In *Proceedings of the First Workshop on Scholarly Document Processing*, pages 251–260, Online. Association for Computational Linguistics.

Jade Goldstein, Vibhu O Mittal, Jaime G Carbonell, and Mark Kantrowitz. 2000. Multi-document summarization by sentence extraction. In *NAACL-ANLP 2000 Workshop: Automatic Summarization*.

Jiaxin Ju, Ming Liu, Longxiang Gao, and Shirui Pan. 2020. Monash-summ@LongSumm 20 SciSummPip: An unsupervised scientific paper summarization pipeline. In *Proceedings of the First Workshop on Scholarly Document Processing*, pages 318–327, Online. Association for Computational Linguistics.

Guy Lev, Michal Shmueli-Scheuer, Jonathan Herzig, Achiya Jerbi, and David Konopnicki. 2019. Talksumm: A dataset and scalable annotation method for scientific paper summarization based on conference talks.

Lei Li, Yang Xie, Wei Liu, Yinan Liu, Yafei Jiang, Siya Qi, and Xingyuan Li. 2020. CIST@CL-SciSumm 2020, LongSumm 2020: Automatic scientific document summarization. In *Proceedings of the First Workshop on Scholarly Document Processing*, pages 225–234, Online. Association for Computational Linguistics.

Peng Li, Yinglin Wang, Wei Gao, and Jing Jiang. 2011. Generating aspect-oriented multi-document summarization with event-aspect model. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yuanhua Lv and ChengXiang Zhai. 2011. Lower-bounding term frequency normalization. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, page 7–16, New York, NY, USA. Association for Computing Machinery.

Chirantana Mallick, Ajit Kumar Das, Madhurima Dutta, Asit Kumar Das, and Apurba Sarkar. 2019. Graph-based text summarization using modified textrank. In *Soft computing in data analytics*, pages 137–146. Springer.

Mani Maybury. 1999. *Advances in automatic text summarization*. MIT press.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

Santosh Kumar Mishra, Harshavardhan Kundarapu, Naveen Saini, Sriparna Saha, and Pushpak Bhattacharyya. 2020. IITP-AI-NLP-ML@ CL-SciSumm 2020, CL-LaySumm 2020, LongSumm 2020. In *Proceedings of the First Workshop on Scholarly Document Processing*, pages 270–276, Online. Association for Computational Linguistics.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Ramesh Nallapati, Bowen Zhou, and Mingbo Ma. 2016. Classify or select: Neural architectures for extractive document summarization. *arXiv preprint arXiv:1611.04244*.

Saichethan Reddy, Naveen Saini, Sriparna Saha, and Pushpak Bhattacharyya. 2020. IIITBH-IITP@CL-SciSumm20, CL-LaySumm20, LongSumm20. In *Proceedings of the First Workshop on Scholarly Document Processing*, pages 242–250, Online. Association for Computational Linguistics.

S. Robertson, S. Walker, Susan Jones, M. Hancock-Beaulieu, and Mike Gatford. 1994. Okapi at trec-3. In *TREC*.

Sajad Sotudeh Gharebagh, Arman Cohan, and Nazli Goharian. 2020. GUIR @ LongSumm 2020: Learning to generate long summaries from scientific documents. In *Proceedings of the First Workshop on Scholarly Document Processing*, pages 356–361, Online. Association for Computational Linguistics.