IITP-AI-NLP-ML@ CL-SciSumm 2020, CL-LaySumm 2020, LongSumm 2020

Santosh Kumar Mishra, Kundarapu Harshavardhan , Naveen Saini, Sriparna Saha and Pushpak Bhattacharyya
Department of Computer Science & Engineering
Indian Institute of Technology Patna
Patna, Bihar, India-801106
{santosh_1821cs03, 1801cs29, naveen.pcs16, sriparna, pb}@iitp.ac.in

Abstract

The publication rate of scientific literature increases rapidly, which poses a challenge for researchers to keep themselves updated with new state-of-the-art. Scientific document summarization solves this problem by summarizing the essential fact and findings of the document. In the current paper, we present the participation of IITP-AI-NLP-ML team in three shared tasks, namely, CL-SciSumm 2020, LaySumm 2020, LongSumm 2020, which aims to generate medium, lay, and long summaries of the scientific articles, respectively. To solve CL-SciSumm 2020 and LongSumm 2020 tasks, three well-known clustering techniques are used, and then various sentence scoring functions, including textual entailment, are used to extract the sentences from each cluster for a summary generation. For LaySumm 2020, an encoder-decoder based deep learning model has been utilized. Performances of our developed systems are evaluated in terms of ROUGE measures on the associated datasets with the shared task.

1 Introduction

Massive amounts of scientific articles are published day by day (Cohan et al., 2015; Cohan and Goharian, 2017, 2018), which impose a big challenge for researchers in various fields to keep themselves up-to-date with the new developments. A bibliometric analyst's study shows that after nine years, the number of published articles will be doubled (Bornmann and Mutz, 2015). The scientific document summarization objective is to provide a summary of the reference paper. This summary should contain all the important facts. Therefore, it reduces the human effort to understand the document.

Challenges of each style of the summary are as follows:

- Objective of CL-SciSumm is to generate a short summary of a paper which must contain all relevant facts and findings. To solve this problem, we have used the extractive summarization technique. We have used unsupervised techniques and explored different features for scientific summarization. These used features help in identifying important sentences of the article using different aspects.

- Objective of LongSumm is to generate a long summary of the scientific article that should be extractive and abstractive. The generated long summary must contain all important facts of the article. To solve the extractive long summarization problem, we have an unsupervised technique similar to CL-SciSumm. To solve the abstractive LongSumm problem, we have used the encoder-decoder based generative model.

- Objective of CL-LaySumm is to generate a lay summary that can be understood by a non-technical reader. The generated summary should not contain any technical words or jargon. We have solved this task using the abstractive summarization technique. Here, Fine-tuned BERT based encoder-decoder architecture is used to solve the problem.

The current paper addresses this issue by participating in the three shared tasks, namely, CL-SciSumm 2020, LaySumm 2020, LongSumm 2020 (Chandrasekaran et al., 2020). These tasks' goals are to generate medium, Lay (understandable for the non-technical audience), and long summaries of the scientific articles. We are using an extractive approach for CL-SciSumm, For LongSumm, an extractive followed by an abstractive approach is utilized, and for LaySumm, an abstractive approach is utilized. The detailed descriptions of these tasks are provided in the subsequent sections.

Proceedings of the First Workshop on Scholarly Document Processing, pages 270–276

https://doi.org/10.18653/v1/P17

2 CL-SciSumm 2020

It is the sixth shared task on scientific document summarization. In the literature, two approaches have been used to solve this problem. The first one considers abstract as the summary of the paper, but the problem with the approach is it provides only the theme of the paper. The abstract may not convey all the important points of the summary (Yasunaga et al., 2019; Atanassova et al., 2016). Therefore, the second approach has been followed to solve the scientific summarization, which is citation-based summarization (Qazvinian et al., 2013). It utilizes a set of citations referencing to the original article (reference paper to be summarized). Citations are short descriptions that explain the reference paper all its contributions; this text can be termed as citation text or citance.

2.1 Dataset

The dataset contains a blind test set of 20 papers and corresponding citing papers. Each paper belongs to the computation linguistics and natural language processing domain. It can be found at https://github.com/WING-NUS/scisumm-corpus/tree/master/data/Test-Set-2018. Training data is also provided. But, as our approach is purely unsupervised; therefore, we are making use of only test data.

2.2 Tasks Descriptions

- **Task-1 (A)** In this task, the objective is to identify the spans of text (cited text spans) in the reference paper (RP) for each citance given the RP and citing papers (CPs).
- **Task-1 (B)** is to classify each cited text span into facets that are predefined (Hypothesis, Aim, Method, Results, and Implication).
- **Task-2** is to produce a summary of the reference paper by utilizing its citation. The generated summary length should be less than or equal to 250 words.

2.3 System Description

In this section, the steps followed in our proposed framework for solving different sub-tasks are elaborated.

Task 1 (A) To find out the cited text span in the reference paper for each citance, we have utilized the word mover's distance (Kusner et al., 2015). For each citation sentence, WMD is used to identify the most similar sentences from the reference paper. WMD denotes the semantic similarity between sentences. Here we have selected the top five most similar sentences from the reference paper.

Task 1 (B) To classify each cited text span, we have calculated the similarity between the cited text span and all the five facets using word mover's distance (WMD). The cited text span is assigned that facet, which is closest in terms of WMD.

Task 2 To generate a structured summary, we have used the unsupervised technique, i.e., clustering, followed by the sentence extraction from each cluster based on various sentence-scoring functions. The sentence having a high score from each cluster is included in the summary until the desired length of the summary is reached. A series of steps followed are as follows:

1. Grouping of the sentences has been done using the traditional clustering techniques, namely, K-means (Lloyd, 1982), K-medoid (Kaufman et al., 1987), and DB-scan (Ester et al., 1996).

2. We have determined the document center/representative sentence (RS) of the reference paper. It is that sentence in the article which is most similar to the remaining sentences. We can also call it as an article's center. In other words, the sentence having the minimum average WMD with respect to other sentences is called the RS.

3. Clusters are ranked based on their distances from the representative sentence (RS). In other words, the cluster closest to the RS is assigned the highest rank.

4. After ranking the clusters, we have calculated the scores of the sentences within each cluster based on the following features and then selected the highest scored sentence from each cluster considering their rankings. Note that the selection of sentences from the ranked clusters (in a sequence) will continue until we get the desired length of the summary.

   Position of the Sentence (F1): In the literature, it has been shown that important sentences are found in the title and lead sentences of a paragraph. It is expressed as follows

$m_i = \sqrt{\frac{1}{n_i}}$ where n_i is the position of a sentence in the reference paper. The sentence is given the highest priority, which lies at the start of the document (Saini et al., 2019).

Similarity with the title (F2): In any document, the sentence, which is very much similar to the title of the document, can be an important sentence for the summary (Saini et al., 2019) as it represents the theme of the article. Here word mover's distance is used to find the similarity.

Length of the Sentence (F3): In the previous works, it is shown that longer sentences can be relevant for generating a summary for a document describing some news (Mendoza et al., 2014; Saini et al., 2019). The sentence is assigned the highest priority, which has the longest length.

Textual Entailment (F4): Textual entailment (Saini et al., 2020) has been used as an anti-redundancy measure. In a good summary, sentences should not be related to each other to have more coverage. Here, initially, the cluster centers are included in the summary following the ranked order of the clusters. In the next step, those sentences are selected from the ranked clusters and included in the summary, which does not entail any sentence in summary.

2.4 Submitted Run

Details of the submitted systems are provided in Table 1. Here we have used five clusters for K-means and K-medoid, whereas DB-scan decides the number of clusters automatically. In Table 1 each run describes the features used for the selection of sentences within clusters to form the summary. Here, twelve different runs have been used for task-2.

2.5 Result

Results of task-1 (a), task-1 (b) and task-2 are shown in Table 2, Table 3 and Table 4 respectively.

	DB-Scan	K-means	K-medoid
F1	run1	run2	run3
F2	run4	run5	run6
F3	run7	run8	run9
F4	run10	run11	run12

Table 1: Details of submitted runs of CL-SciSumm

3 CL-LaySumm 2020

The motivation of the CL-LaySumm Shared Task is to automatically produce Lay Summaries of technical (scientific research article) texts. A Lay Summary is defined as a textual summary easily understood by a non-technical audience. It is typically produced either by the authors or by a journalist or commentator. The corpus released in shared tasks covers three distinct domains: epilepsy, archeology, and materials engineering.

In a lay summary, there should not be any technical jargon. It should reflect the overall scope, goal, and potential impact of a scientific paper. It is typically less than 150 words in length. The objective is to generate summaries that represent the content, understandable and interesting to a lay audience.

3.1 Dataset

The dataset has a training set of 572 articles having corresponding lay summaries. It contains a blind test set of 37 papers.

3.2 System Description

Neural network based approach formulates abstractive summarization problem as sequence to sequence problem, here encoder is used to read the token of source documents $\mathbf{x} = [x_1, x_2, ...x_n]$ into an intermediate representation $\mathbf{z} = z_1, z_2, ...z_n$. Finally, decoder uses the intermediate representation to generate the final summary $\mathbf{y} = y_1, y_2, ...y_m$ token by token by using conditional probability $p(y_1, ...y_m|x_1, ...x_n)$.

We have used standard encoder-decoder architecture for our lay summarization task. Here the encoder is pre-trained BERTSUM, and it is fine-tuned on CNN daily mail dataset, and the decoder is a six-layer transformer network (as shown in Fig 1). It should be noted that there is a difference between encoder and decoder as the encoder is pre-trained while the decoder has to be trained. This can create an unstable process of fine-tuning, due to which encoder and decoder can have the problem of under-fitting and over-fitting. To resolve this problem, the different optimizers for encoder and decoder have been used.

Here two Adam optimizers are used with $\beta_1 = 0.9$ and $\beta_2 = 0.99$, respectively, along with different learning rate and warm-up states as follows:

$$l_{r\epsilon} = \tilde{l}_{r\epsilon}.min(step^{0.5}, step.warmup_{\epsilon}^{-1.5}) \quad (1)$$

Precision		Recall		F1 score	
micro_avg	**macro_avg**	**micro_avg**	**macro_avg**	**micro_avg**	**macro_avg**
0.0222	0.0221	0.1049	0.1058	0.0367	0.0365

Table 2: Scores of Task-1 (a)

Precision		Recall		F1 score	
micro_avg	**macro_avg**	**micro_avg**	**macro_avg**	**micro_avg**	**macro_avg**
0.0169	0.0364	0.0148	0.0162	0.0158	0.0224

Table 3: Scores of Task-1 (b)

Runs	**Human**		**Community**		**Abstract**	
	R-2	**R-SU4**	**R-2**	**R-SU4**	**R-2**	**R-SU4**
Run 1	0.1028	0.0833	0.1482	0.0899	0.0959	0.0622
Run 2	0.1229	0.0893	0.1561	0.0889	0.1377	0.0669
Run 3	0.1154	0.0888	0.1283	0.0733	0.1206	0.0673
Run 4	**0.1749**	**0.1169**	**0.1897**	**0.1208**	**0.1959**	**0.0962**
Run 5	0.1430	0.1002	0.1624	0.0998	0.1649	0.081
Run6	0.1380	0.1121	0.1245	0.0845	0.1508	0.0856
Run7	0.0997	0.0760	0.1768	0.1013	0.1134	0.0627
Run 8	0.1156	0.0746	0.1658	0.0836	0.1104	0.0610
Run 9	0.0992	0.0732	0.1614	0.0765	0.1187	0.0647
Run 10	0.1251	0.0883	0.1605	0.0989	0.1356	0.0671
Run 11	0.1221	0.0883	0.1602	0.0913	0.1274	0.0703
Run 12	0.1217	0.0938	0.1145	0.0713	0.1194	0.0678

Table 4: Task-2 scores of different runs in terms of rouge scores. Here, R-2 and R-SU4 are denoting rouge-1, rouge-SU4 respectively. All the reported values are f1 scores.

$$l_{rD} = \tilde{l}_{rD}.min(step^{0.5}, step.warmup_D^{-1.5}) \quad (2)$$

where $l_{r\epsilon} = 2e^{-}3$ and warm-up = 20000 for the encoder whereas $l_{rD} = 0.1$ and warm-up =10000 for decoder. Here the assumption is that the pre-trained encoder must be trained with a lower learning rate and a lower learning rate smoothens the decay. This process helps the encoder in training with a better gradient when the decoder is in stable condition.

We have used a two-stage fine-tuning approach, first is fine-tuning for extractive summarization and then for abstractive summarization. It has been shown in the literature (Li et al., 2018) (Gehrmann et al., 2018) extractive object helps in obtaining a better abstractive summary.

3.3 Result

Our system has the following score (shown in Table 5).

4 LongSumm 2020

In all the previous works of scientific summarization (Cohan and Goharian, 2017, 2018), there is a summary length constraint of a maximum of 250 words. But in the current LongSumm shared task, the generated summary can be the length of between 100-1500 words.

4.1 Dataset

This dataset consists of a training set of 1705 papers associated with extractive summaries and 531 papers associated with abstractive summaries. It has a blind test set of 22 files. It can be found at https://github.com/guyfe/LongSumm.

4.2 System Description

We have used both extractive and abstractive approaches on the blind test set to generate a structured summary. We have used clustering followed by the sentence-scoring and extraction procedure within each cluster based on various features. The sentence having the highest score is included in the

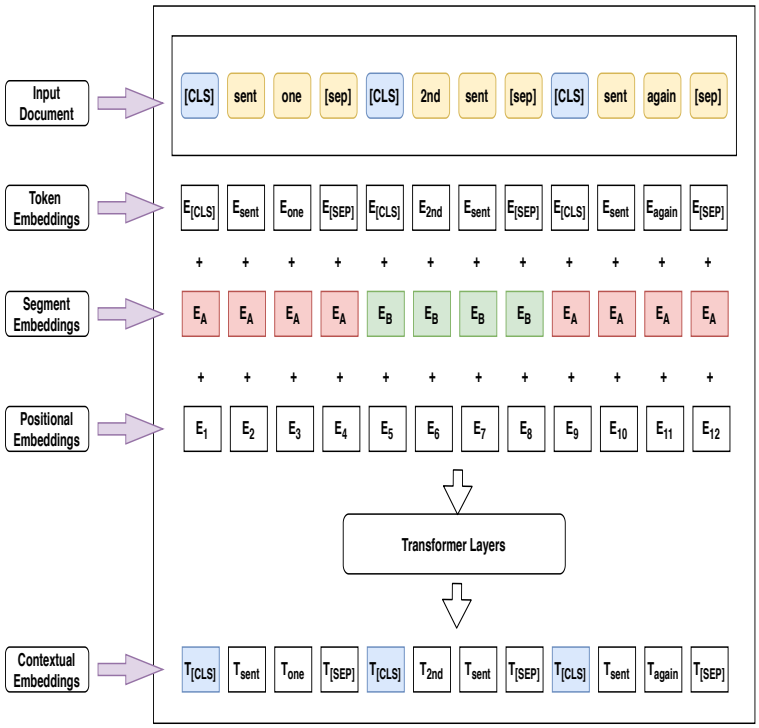

Figure 1: Architecture for Lay Summarization task

R-1 (f)	R-2 (r)	R-2 (f)	R-2 (r)	R-l (f)	R-l (r)
0.3132	0.3705	0.0631	0.0746	0.1662	0.1973

Table 5: Obtained scores on blind set of CL-LaySumm, Here, R-1, R-2, and R-l are denoting rouge-1, rouge-2, and rouge-l, respectively, f and r representing f-1 score and recall, respectively.

summary. Note that for LongSumm extractive summarization, we have used the same methodology as used for CL-SciSumm 2020. A deep learning-based encoder-decoder model is used for LongSumm abstractive summarization.

4.3 Submitted Run:

Details of submitted systems are provided in Table 6. Here we have used five clusters for K-means and K-medoid, whereas DB-scan decides the number of clusters automatically. In Table 6 each run describes the features used for the selection of sentences within the cluster to form the summary.

	DB-Scan	K-means	K-medoid
F1	run1	run2	run3
F2	run4	run5	run6
F3	run7	run8	run9
F4	run10	run11	run12

Table 6: Details of submitted runs of LongSumm

We have used deep learning-based technique as run13, which is as follows:

run13: Proposed model has utilized encoder-decoder based deep learning model. Fine-tuned BERT model has been used for the generation of embedding. We have used same model as used for LaySumm.

4.4 Result

The results of all runs are shown in Table 7, here from run-1 are run-12 are the scores of long summarization using an extractive approach, whereas run-13 is the score of long summarization using the abstractive approach.

5 Conclusion

This paper has presented the results of participation of the IITP-AI-NLP-ML team in three shared tasks, namely, CL-SciSumm 2020, CL-LaySumm 2020, LongSumm 2020, at SDP 2020. For CL-SciSumm, three sub-tasks are there: Task 1(A), Task 1(B), and Task 2. For Task 1 (A), we have utilized WMD

Runs	R-1 (f)	R-1 (r)	R-2 (f)	R-2 (r)	R-l (f)	R-l (r)
Run 1	0.4112	0.4226	0.4112	0.0967	0.1539	0.1581
Run 2	0.4469	0.425	0.4469	0.1128	0.1675	0.1591
Run 3	0.4112	0.4226	0.4112	0.0967	0.1539	0.1581
Run 4	0.3962	0.4062	0.3962	0.094	0.1503	0.1538
Run 5	0.3948	0.3815	0.3948	0.096	0.144	0.1393
Run 6	0.3554	0.3657	0.3554	0.0868	0.1301	0.1337
Run7	0.335	0.3432	0.335	0.0803	0.1283	0.1313
Run 8	0.4485	0.4288	0.4485	0.1099	0.1667	0.1592
Run 9	0.4448	0.4564	0.4448	0.1207	0.1638	0.1677
Run 10	0.4631	0.4723	0.4631	0.1345	0.1749	0.1784
Run 11	0.4597	0.4366	0.4597	0.1368	0.1778	0.1687
Run 12	0.449	0.4603	0.449	0.1385	0.1679	0.1721
Run 13	**0.4646**	**0.4743**	**0.4646**	**0.1486**	**0.1958**	**0.1995**

Table 7: Scores obtained by different runs for LongSumm. Here, R-1, R-2, and R-L are denoting rouge-1, rouge-2, and rouge-l, respectively, f and r representing f-1 score and recall, respectively.

to extract the cited text span from the reference paper; for task 1 (B), the similarity-based measure has been used to identify the facet of each cited text span. Task 2 is based on clustering, followed by sentence extraction from each cluster based on their relevance/score. For LongSumm, we have utilized clustering and deep learning techniques and reported 13 different ways to generate a long summary. For LaySumm, we have proposed a deep learning-based encoder-decoder model that generates the lay summary utilizing the fine-tuned BERT language model's embedding.

Acknowledgments

Special thanks to Mr. Saichethan Miriyala Reddy for helping us at various levels.

References

Iana Atanassova, Marc Bertin, and Vincent Larivière. 2016. On the composition of scientific abstracts. *Journal of Documentation*, 72(4):636–647.

Lutz Bornmann and Rüdiger Mutz. 2015. Growth rates of modern science: A bibliometric analysis based on the number of publications and cited references. *Journal of the Association for Information Science and Technology*, 66(11):2215–2222.

M. K. Chandrasekaran, G. Feigenblat, Hovy. E., A. Ravichander, M. Shmueli-Scheuer, and A De Waard. 2020. Overview and insights from scientific document summarization shared tasks 2020: CL-SciSumm, LaySumm and LongSumm. In *Proceedings of the First Workshop on Scholarly Document Processing (SDP 2020)*.

Arman Cohan and Nazli Goharian. 2017. Scientific article summarization using citation-context and article's discourse structure. *arXiv preprint arXiv:1704.06619*.

Arman Cohan and Nazli Goharian. 2018. Scientific document summarization via citation contextualization and scientific discourse. *International Journal on Digital Libraries*, 19(2-3):287–303.

Arman Cohan, Luca Soldaini, and Nazli Goharian. 2015. Matching citation text and cited spans in biomedical literature: a search-oriented approach. In *proceedings of the 2015 conference of the North American Chapter of the association for computational linguistics: Human language technologies*, pages 1042–1048.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231.

Sebastian Gehrmann, Yuntian Deng, and Alexander M Rush. 2018. Bottom-up abstractive summarization. *arXiv preprint arXiv:1808.10792*.

L Kaufman, PJ Rousseeuw, and Y Dodge. 1987. Clustering by means of medoids in statistical data analysis based on the.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *International conference on machine learning*, pages 957–966.

Wei Li, Xinyan Xiao, Yajuan Lyu, and Yuanzhuo Wang. 2018. Improving neural abstractive document summarization with explicit information selection modeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1787–1796.

Stuart Lloyd. 1982. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137.

Martha Mendoza, Susana Bonilla, Clara Noguera, Carlos Cobos, and Elizabeth León. 2014. Extractive single-document summarization based on genetic operators and guided local search. *Expert Systems with Applications*, 41(9):4158–4169.

Vahed Qazvinian, Dragomir R Radev, Saif M Mohammad, Bonnie Dorr, David Zajic, Michael Whidby, and Taesun Moon. 2013. Generating extractive summaries of scientific paradigms. *Journal of Artificial Intelligence Research*, 46:165–201.

Naveen Saini, Sriparna Saha, Pushpak Bhattacharyya, and Himanshu Tuteja. 2020. Textual entailment–based figure summarization for biomedical articles. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 16(1s):1–24.

Naveen Saini, Sriparna Saha, Dhiraj Chakraborty, and Pushpak Bhattacharyya. 2019. Extractive single document summarization using binary differential evolution: Optimization of different sentence quality measures. *PloS one*, 14(11).

Michihiro Yasunaga, Jungo Kasai, Rui Zhang, Alexander R Fabbri, Irene Li, Dan Friedman, and Dragomir R Radev. 2019. Scisummnet: A large annotated corpus and content-impact models for scientific paper summarization with citation networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7386–7393.