Overview and Insights from the Shared Tasks at Scholarly Document Processing 2020: CL-SciSumm, LaySumm and LongSumm

Muthu Kumar Chandrasekaran
Amazon USA
cmkumar087@gmail.com

Guy Feigenblat
IBM Research AI
guyf@il.ibm.com

Eduard Hovy
Carnegie Mellon University
hovy@cmu.edu

Abhilasha Ravichander
Carnegie Mellon University
aravicha@cs.cmu.edu

Michal Shmueli-Scheuer
IBM Research AI
shmueli@il.ibm.com

Anita de Waard
Elsevier
a.dewaard@elsevier.com

Abstract

We present the results of three Shared Tasks held at the Scholarly Document Processing Workshop at EMNLP2020: CL-SciSumm, LaySumm and LongSumm. We report on each of the tasks, which received 18 submissions in total, with some submissions addressing two or three of the tasks. In summary, the quality and quantity of the submissions show that there is ample interest in scholarly document summarization, and the state of the art in this domain is at a midway point between being an impossible task and one that is fully resolved.

1 Introduction

Scientific documents constitute a rich field for different tasks such as Reference String Parsing, Citation Intent Classification, Summarization and more. The constantly increasing number of scientific publications raises additional issues such as making these publications accessible to non-expert readers, or, on the other hand, to experts who are interested in a deeper understanding of the paper without reading a paper in full.

For this year's Scholarly Document Processing workshop (Chandrasekaran et al., 2020) at EMNLP 2020, we proposed three tasks: *CL-SciSumm*, *LaySumm* and *LongSumm* to improve the state of the art for different aspects of scientific document summarization.

The *CL-SciSumm* task was introduced in 2014 and aims to explore the summarization of scientific research in the domain of computational linguistics research. It encourages the incorporation of new kinds of information in automatic scientific paper summarization, such as the facets of research information being summarized in the research paper. CL-SciSumm also encourages the use of citing mini-summaries written in other papers, by other scholars, when they refer to the paper.

LaySumm (Lay Summarization) addresses the issue of making research results available to a larger audience by automatically generating 'Lay Summaries', or summaries that explain the science contained within the paper in laymen's terms.

Finally, the *LongSumm* (Long Scientific Document Summarization) task focuses on generating long summaries of scientific text. It is fundamentally different than generating short summaries that mostly aim at teasing the reader. The LongSumm task strives to learn how to cover the salient information conveyed in a given scientific document, taking into account the characteristics and the structure of the text. The motivation for LongSumm was first demonstrated by the IBM Science Summarizer system, (Erera et al., 2019) that retrieves and creates long summaries of scientific documents[1]. While Erera et al. (2019) studied some use-cases and proposed a summarization approach with some human evaluation, the authors stressed the need of a large dataset that will unleash the research in this domain. *LongSumm* aims at filling this gap by providing large dataset of long summaries which are based on blogs written by Machine Learning and NLP experts.

In this paper we present the tasks, datasets, description of the participating systems, and provide their results and insights from shared tasks.

2 CL-SciSumm

2.1 Overview

The CL-SciSumm Shared Task was launched in 2014 as a pilot task aimed at bringing together the summarization community to address challenges in scientific communication summarization. Over time, the Shared Task has spurred the creation

[1] https://ibm.biz/sciencesum

Proceedings of the First Workshop on Scholarly Document Processing, pages 214–224
Online, November 19, 2020. ©2020 Association for Computational Linguistics
https://doi.org/10.18653/v1/P17

of new resources (e.g., (Yasunaga et al., 2019)), tools and evaluation frameworks. As a consequence of this wide interest, CL-SciSumm 2020 is jointly organised with the inaugural editions of two other Scientific Summarization shared tasks, all of which were held as part of SDP 2020 workshop at EMNLP[2]) (Chandrasekaran et al., 2020)

A pilot CL-SciSumm task was conducted at TAC 2014, as part of the larger BioMedSumm Task[3]. In 2016, a second CL-Scisumm Shared Task (Jaidka et al., 2018) was held as part of the Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL) workshop at the Joint Conference on Digital Libraries (JCDL 2016). From 2017 (Jaidka et al., 2017, 2019) through 2019 (Chandrasekaran et al., 2019) CL-SciSumm was colocated with BIRNDL at the annual ACM Conference on Research and Development in Information Retrieval (ACM SIGIR 2017–2019).

In this section we provide the results and insights from CL-SciSumm 2020.

2.1.1 Corpus

We built the CL-SciSumm corpus by randomly sampling research papers (Reference papers, RPs) from the ACL Anthology corpus and then downloading the citing papers (CPs) for those which had at least ten citations. The prepared dataset then comprised annotated citing sentences for a research paper, mapped to the sentences in the RP which they referenced. Summaries of the RP were also included.

The CL-SciSumm 2020 corpus consisted of 40 annotated RPs and their CPs. These are the same as described in our overview paper in CL-SciSumm 2019 (Chandrasekaran et al., 2019) and 2018. The test set was blind. We reused the blind test we used from CL-SciSumm 2018 and 2019 since we want to have a comparable evaluation CL-SciSumm 2020 systems. After 3 iterations, we now release the gold labels for the 2018 test-set.

For details of the general procedure followed to construct the CL-SciSumm corpus, and changes made to the procedure in CL-SciSumm-2016, please see (Jaidka et al., 2018). In 2017, we made revisions to the corpus to remove citances from passing citations. These are described in (Jaidka et al., 2017).

Annotation. Given each RP and its associated CPs, the annotation group was instructed to find citations to the RP in each CP. Specifically, the citation text, citation marker, reference text, and discourse facet were identified for each citation of the RP found in the CP. The corpus has 40 annotated RPs, exclusive of 1000 auto-annotated RPs added in CL-SciSumm 2019. For CL-SciSumm-20 we encourage participants to use out-of-domain data (i.e., scientific document corpora from papers outside of the ACL anthology corpora; e.g., BIGPATENT (Sharma et al., 2019)) to bootstrap training using transfer learning. From 2019 onward, Task 2, training data (summaries) has been augmented with the SciSummNet corpus (Yasunaga et al., 2019).

2.1.2 Task

CL-SciSumm defined two serially dependent tasks that participants could attempt, given a canonical training and testing set of papers.

Given: A topic consists of a Reference Paper (RP) and ten or more Citing Papers (CPs) that all contain citations to the RP. In each CP, the text spans (i.e., citances) have been identified that pertain to a particular citation to the RP. Additionally, the dataset provides three types of summaries for each RP:

- the abstract, written by the authors of the research paper.
- the community summary, collated from the reference spans of its citances.
- a human-written summary, written by the annotators of the CL-SciSumm annotation effort.

Task 1A: For each citance, identify the spans of text (cited text spans) in the RP that most accurately reflect the citance. These are of the granularity of a sentence fragment, a full sentence, or several consecutive sentences (no more than 5).

Task 1B: For each cited text span, identify what facet of the paper it belongs to, from a predefined set of facets.

Task 2: Finally, generate a structured summary of the RP from the cited text spans of the RP. The length of the summary should not exceed 250 words. This was an optional bonus task.

2.1.3 Evaluation

An automatic evaluation script was used to measure system performance for **Task 1A**, in terms of the sentence ID overlaps between the sentences identified in system output, versus the gold standard created by human annotators. The raw number

[2] https://2020.emnlp.org/

[3] http://www.nist.gov/tac/2014

of overlapping sentences were used to calculate the precision, recall and F_1 score for each system. We followed the approach in most SemEval tasks in reporting the overall system performance as its micro-averaged performance over all topics in the blind test set.

Additionally, we calculated lexical overlaps in terms of the ROUGE-2 scores (Lin, 2004) between the system output and the human annotated gold standard reference spans.

We have been reporting ROUGE scoring since CL-SciSumm 17, for Tasks 1a and Task 2.

Task 1B was evaluated as a proportion of the correctly classified discourse facets by the system, contingent on the expected response of Task 1A. As it is a multi-label classification, this task was also scored based on the precision, recall and F_1 scores.

Task 2 was optional, and also evaluated using the ROUGE–2 between the system output and three types of gold standard summaries of the research paper: the reference paper's abstract, a community summary, and a human summary.

We provisioned the evaluation scripts and gold-test-set CL-SciSumm Github repository[4]. For transparency we published all the system runs submitted by the participants. The participants then ran the evaluation and reported the results back to us. We collate and publish these as the CL-SciSumm'20 official result.

2.2 Systems Overview

Following teams submitted systems for evaluation for Task 1a and 1b. Their systems are described in their cited systems papers: NJUST (Zhang et al., 2020), CIST (Li et al., 2020), AUTH (Gidiotis et al., 2020), CiteQA (Umapathy et al., 2020), IITBH-IITP (Reddy et al., 2020), IITP-AI-NLP-ML (Mishra et al., 2020), MLU (Huang and Krylova, 2020), MLUHW (Boltze et al., 2020), UniHD (Aumiller et al., 2020), NLP-PINGAN-TECH (Chai et al., 2020)
Following teams submitted systems for evaluation on Task 2 also which is an optional bonus task: AUTH (Gidiotis et al., 2020), CIST (Li et al., 2020), IITBH-IITP (Reddy et al., 2020), IITP-AI-NLP-ML (Mishra et al., 2020)

Official evaluation results on these systems is presented in the next section.

2.3 Results

Out of the 11 participants systems, 8 were able complete the final evaluation correctly. We have excluded the rest 3 them from listing in Tables 1 and 2 in the results on the blind test set. However, their systems and results on the development set are published in their respective system papers. We allows teams to submit an unlimited number of runs since this is an offline evaluation with a blind test set. However, we tabulate only the results from the top 5 runs when a large of runs are submitted.

Task 1a. (Table 1)NLP-PINGAN-TECH(Chai et al., 2020) achieve the best result on Task 1a when evaluated using sentence overlaps and ngram overlaps using ROUGE SU4. All top 5 of their runs outperforms other systems. Runs from UniHD's system are a close second.

Task 1b. (Table 2) We note that the runs that perform the best on Task1a are not the same that top performance in Task 1b though Task 1b is evaluated conditioned on Task 1a. CIST (Li et al., 2020)'s systems do consistently well on this task. We note that UniHD's systems, intersection_2_field and intersection_3_field do well on both Task 1a and 1b though they do not top the rankings on either task.

Task 2. Four of the eleven teams also particiapted in the bonus summarization task. On the summarization task AUTH (Gidiotis et al., 2020) does well when evaluated against both abstract and human written summaries. They score 0.41 on ROUGE-2 on Abstracts which is comparable to the state-of-the-art of general summarization. However, their system does not do well on community summaries, which is dependant on Task 1a. IITBH-IITP (Reddy et al., 2020)'s systems consistently perform better than the rest on community summaries. CIST (Li et al., 2020)'s systems are second and are comparable to the top performing system in this category. Notably CIST's runs do well on both human and community summaries and second only to AUTH on abstracts. This type of systems are the intended goal of the CL-SciSumm shared task.

3 LaySumm

3.1 Task Overview

To improve public understanding of science, researchers are increasingly asked by funders and publishers to outline the scope of their research, described in scientific research articles, by writing a summary for a lay audience. We call this a

[4] github.com/WING-NUS/scisumm-corpus

System	Task 1A: Sentence Overlap (F_1)	Task 1A: ROUGE-SU4 F_1	System	Task 1B (F_1)
NLP_PINGAN_TECH sembert_scibert_all_top2	**0.17**	**0.15**	CIST run40	**0.41**
NLP_PINGAN_TECH run_scibert_unused_token_top2	**0.17**	**0.15**	CIST run42	**0.41**
NLP_PINGAN_TECH sembert_sembert_scibert_all_top3	0.17	0.11	CIST run41	**0.41**
NLP_PINGAN_TECH run_scibert_all_top3	0.17	0.10	CIST run61	0.39
NLP_PINGAN_TECH run_scibert_all_top2	0.17	0.14	CIST run62	0.39
uniHD intersection_2_field	0.16	0.11	CMU run26	0.31
uniHD intersection_3_field	0.15	0.08	CMU run27	0.31
CMU run110	0.13	0.08	CMU run110	0.30
CMU run12	0.13	0.09	uniHD intersection_3_field	0.29
CMU run13	0.13	0.09	uniHD intersection_2_field	0.29
CMU run32, 33	0.13	0.11	CMU run24, 25	0.29
uniHD negative_only_2_field	0.12	0.06	NLP_PINGAN_TECH run_sembert_scibert_all_top3	0.23
uniHD with_truth_2_field	0.12	0.06	NLP_PINGAN TECH run_scibert_all_top3	0.23
uniHD negative_only_3_field	0.12	0.06	IITBH-IITP variantU	0.23
CIST runs 22-42	0.11	0.05	NLP_PINGAN TECH run_scibert_2_top3	0.21
uniHD with_truth_3_field	0.11	0.05	NLP_PINGAN TECH run_sembert_top3	0.21
CIST runs 43-63	0.11	0.05	NLP_PINGAN TECH run_only_scibert_sp_token_top3	0.21
AUTH run 2	0.10	0.09	AUTH run 1	0.17
IITBH-IITP variantU	0.08	0.03	IITBH-IITP variantF	0.16
IITBH-IITP variantF	0.06	0.03	IITBH-IITP variantA	0.13
CIST runs 1-21	0.05	0.10	IITBH-IITP variantE	0.08
CIST runs 67-72	0.05	0.09	IITBH-IITP variantS	0.06
CIST runs 64-66,73-84	0.05	0.09	IITP-AI-NLP-ML	0.02
IITP-AI-NLP-ML runs 1-10	0.04	0.01	MLU Halle-Wittenberg	0.01
IITBH-IITP variantA	0.03	0.01	IITBH-IITP variantX	0.01
IITBH-IITP variantE	0.02	0.01		
IITBH-IITP variantS	0.02	0.01		
MLU Halle-Wittenberg	0.01	0.02		

Table 1: CL-SciSumm systems' performance in Task 1A and 1B, ordered by their F_1-scores for sentence overlap on Task 1A, Task 1B separately. Each system's rank by their performance on ROUGE on Task 1A is shown in parentheses.

Lay Summary: a text of about 70–100 words intended for a non-technical audience that explains, succinctly and without using technical jargon, the overall scope, goal, and potential impact expressed in a scientific paper. The Lay Summarization task provides data for and evaluates automatically-produced Lay Summaries.

3.1.1 Corpus

The corpus comprised 572 author-generated lay summaries from a multidisciplinary collection of journals in Materials Science, Archaeology, Hepatology and Artificial intelligence, together with their corresponding abstracts and full text articles, provided by Elsevier. A small sample dataset can be found on the GitHub repository[5]). A training corpus of 37 full-text papers and abstracts was made available to enable evaluation.

3.1.2 Task

The Lay Summary Task requires systems to generate a lay summary, given a full-text paper and its abstract. This summary should be representative of the content, comprehensible, and interesting to a lay audience. In addition to their results, system builders were asked to provide an automatically generated lay summary of their own system-description paper. The task was run on CodaLabs[6].

3.1.3 Evaluation

We measured summary quality using the ROUGE measure (Lin, 2004). We used the *Py-Rouge* 0.1.3 package, which is built on the ROUGE 1.5.5 toolkit with its standard parameters setting[7]. We report both Recall and F-Measure for ROUGE-1, ROUGE-2, and ROUGE-L. The evaluation results were displayed on a public leaderboard on Codalab[8]. In addition, a number of automatically

[5] https://github.com/WING-NUS/scisumm-corpus/blob/master/README_Laysumm.md#sample-dataset

[6] https://competitions.codalab.org/competitions/25516#learn_the_details

[7] ROUGE-1.5.5.pl -a -c 95 -m -n 2 -2 4 -u -p 0.5

[8] https://competitions.codalab.org/competitions/25516

System	Abstract	Community	Human
	R–2	R–2	R–2
AUTH run 2 2	**0.41**	0.11	**0.22 (1)**
CIST run43, 46, 49 52, 55, 58, 61	0.21	0.24(4)	0.18(4)
CIST run22, 25, 28 31, 34, 37, 40	0.20	0.25(3)	0.20(2)
CIST run 1, 10, 13 16, 19, 4, 7	0.20	0.22	0.19(3)
IIT-NLP-AI-ML run 4	0.20	0.19	0.17(6)
CIST run 64, 67 70, 73, 76, 79, 82	0.18	0.23(6)	0.18(4)
IIT-NLP-AI-ML run5	0.16	0.16	0.14
IIT-NLP-AI-ML run6	0.15	0.12	0.14
IITBH-IITP variant A2 E2, F2, S2, U2, X2	0.15	0.14	0.15
CIST run 11,14 17, 2, 20, 5, 8	0.14	0.15	0.14
IIT-NLP-AI-ML run2	0.14	0.16	0.12
IIT-NLP-AI-ML run10	0.14	0.16	0.13
CIST run 45, 48, 51 54, 57, 60, 63	0.12	0.18	0.13
CIST run 24, 27 30, 33, 36, 39, 42	0.12	0.17	0.16
IIT-NLP-AI-ML run7	0.11	0.18	0.10
IIT-NLP-AI-ML run8	0.11	0.17	0.12
IIT-NLP-AI-ML run9	0.11	0.16	0.10
IITBH-IITP variantU	0.10	**0.27(1)**	0.13
IITBH-IITP variantF	0.09	0.26(2)	0.11
IITBH-IITP variantA	0.07	0.24(4)	0.10
IITBH-IITP variantE	0.09	0.23(6)	0.11
IITBH-IITP variantS	0.13	0.19	0.14
IITBH-IITP variantX	0.06	0.17	0.09

Table 2: CL-SciSumm systems' performance for Task 2 ordered by their ROUGE–2(R–2) F_1-scores. Systems' rank by their performance on the corresponding evaluation is shown in parentheses for the top 5 scores in that category. Winning scores are bolded.

generated lay summaries underwent human evaluation by science journalists and communicators for comprehensiveness, legibility, and interest.

3.2 Systems Overview

We received eight submissions. We briefly describe the approaches taken by the participating teams:
AUTH (Gidiotis et al., 2020) – The authors use a summarization method utilizing PEGASUS (Zhang et al., 2019) to compress and rewrite the abstract of a given article to generate a lay summary. The PEGASUS model is fine-tuned to generate lay summaries, using the article abstract as input and the lay summary as the reference for training the summarization model.
Dimsum (Tiezheng Yu and Fung, 2020) - The system generates a summary by using a joint extractive and abstractive summarization approach, based on the intuition that lay summaries are grounded in sentences that occur within the scientific document. The abstractive summaries are converted to extractive labels, by selecting sentences that maximize the rouge score with the reference summary. The BART encoder (Lewis et al., 2020) is then used to make sentence representations and the model is trained with both extractive and abstractive summarization objectives.
Seungwon (Kim, 2020) - The system built by the team from Georgia Tech primarily uses the PEGASUS model (Zhang et al., 2019) to generate lay summaries, combining this with a BERT-based extractive summarization model. After generating a lay summary using PEGASUS, if the generated summary is shorter than a specified length, the extractive model is used to identify candidate sentences in the document that can be included in the summary. Sentences are only included in the summary by the extractive model if they are judged sufficiently readable, according to a sentence readability metric defined by the authors.
IIITBH-IITP (Reddy et al., 2020) - The authors use an extractive sentence classification method. They develop an unsupervised approach, selecting sentences from the document using variants of the maximum marginal relevance (MMR) metric.
Summaformers (Roy et al., 2020) - This system utilizes the BART model (Lewis et al., 2020) to generate summaries. BART is trained on the CNN/Dailymail summarization dataset (See et al., 2017) and fine-tuned on the Laysumm corpus.
IITP-AI-NLP-ML (Mishra et al., 2020) This method uses a standard encoder-decoder framework for abstractive summarization. The system is based on BERT fine-tuned on the CNN/Dailymail dataset (Liu and Lapata, 2019a), with a decoder consisting of six transformer layers.
DUCS: (no paper submitted) This system uses a two-stage pipeline. In the first phase, extractive summarization is performed, and relevant sentences are selected from the introduction, discussion and conclusion of the article. The abstract, and the extracted sentences from the introduction, discussion and conclusion are summarized using the BART model (Lewis et al., 2020), and the summaries are concatenated.

3.3 Results

Taking these metrics into account, the top 3 systems are: #1 Seungwon Kim, #2 HYTZ, and #3

Table 3: ROUGE Recall and F-Measure evaluation on LaySumm test set

System	Rouge1-F1	Rouge1-Recall	Rouge2-F1	Rouge2-Recall	RougeL-F1	RougeL-Recall
HYTZ	0.4600	0.5013	0.2070	0.2223	0.2876	0.3104
seungwonkim	0.4596	0.4810	0.2146	0.2237	0.2977	0.3105
Summaformers	0.4594	0.4911	0.1902	0.2026	0.2744	0.2923
AUTH	0.4456	0.4298	0.1936	0.1860	0.2772	0.2673
DUCS	0.4253	0.5159	0.1748	0.2102	0.2526	0.3055
IIITBH-IITP	0.4048	0.5414	0.1690	0.2253	0.2244	0.3019
Harita_ramesh_babu	0.3524	0.3865	0.1110	0.1232	0.1995	0.2188
IITP-AI-NLP-ML	0.3132	0.3705	0.0631	0.0746	0.1662	0.1973

Summaformers. Next to the formal ROUGE scores, a subset of documents was evaluated by a team of domain experts. Gratifyingly, this human assessment confirmed this order of the results. Overall, the majority of submitted Lay Summaries was easy to read, though in some cases there were odd errors (e.g., inserted ellipses). The winning systems all produced legible and accessible summaries.

Four of the papers complied with the request that the systems generate a Lay Summary of their own paper, using their own tools. This helps both to explain the concept of a Lay Summary and offers insights into the output of the software; hopefully it also helps explain this work to a non-specialised audience. For examples, please see the Lay Summary Submissions elsewhere in this Anthology.

3.4 Discussion

A comparison of Lay Summaries against typical paper abstracts (Technical Summaries) reveals several systematic differences. These include:

- Lexical specialization: This category includes both domain-based terminological difference (e.g., “renal” vs “kidney” failure, “high-octane” vs “powerful” gasoline) and conceptual specificity / specialization (e.g., “bubblesort” vs “sorting”, “kNN” vs “clustering”). Used at even the same level of specificity, the expert uses domain-specialist words. It is well known that experts’ Basic Level categories (in the sense of Prototype Theory) (Rosch, 1973) is one level lower/more specific than normal speakers’ categories.
- Syntactic complexity: This includes more-complex descriptive NPs vs simpler NPs across more sentences, and longer and deeper sentence parse trees vs shorter and more straightforward ones. Generally an expert author’s abstract has no direct verb forms and no personal pronouns, while the lay summary has nothing but. Direct quotes typically make a lay summary read like journalism.
- Epistemic complexity: Expert text includes more (and more-precise) hedging vs simper, more absolutist claims, and fewer evaluative interjections (“surprising”, “lovely”, “elegant”).
- Content detail: Generally a lay content is more general, wider-ranging, and includes a historically longer but much shallower historical overview compared to the Related Work section of an expert text. Typically there are more examples in the lay text and the examples employ out-of-domain scenarios/entities.
- Author presence: In lay summaries there is generally more explicit ‘author foregrounding’, leading to the personalization of the knowledge source. The opposite in expert summaries has been argued as suggesting there statement of known facts, a tactic that scientists often use.

As described in the previous section, only a few systems implemented some of these strategies explicitly. Generally the hope was that the training data will allow a sufficiently powerful machine learning model to learn what to do by itself. The results do not really bear out this hope. We believe there is some very interesting and fruitful analysis to be done in order to create machine-learning models that are sufficiently rich to produce truly interesting and readable Lay Summaries.

4 LongSumm

4.1 Task Overview

Existing work on scientific document summarization focuses on generating short, abstract-like summaries. While this might be appropriate when summarizing news articles, such summaries cannot cover all the salient information conveyed in a scientific paper. Writing longer summaries requires deep understanding and domain expertise, as can be found in research blogs. To address this point, the LongSumm task opted to leverage blog posts created by researchers in the NLP and Machine learning communities that summarize scientific articles and use these posts as reference summaries (Boni et al., 2020). The task is, given a scientific document, generate a 600 words summary.

4.1.1 Corpus

The 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 from associated conferences (Lev et al., 2019), and contain up to 30 sentences. The abstractive summaries are blog posts created by NLP and ML researchers, with length varied between 100-1500 words, an average of 779 (±460) words, and an average of 31 (±18) sentences in a summary. In addition, we created a (blind) test set of 22 abstractive summaries for evaluating the submissions. The corpus can be found on LongSumm GitHub repository[9].

4.1.2 Evaluation

We measured summarization quality using the ROUGE measure (Lin, 2004). The evaluation script utilizes the *rouge-score*[10] python package which is designed to replicate results from the original perl package with its standard parameters. We report both Recall and F-Measure of ROUGE-1, ROUGE-2, and ROUGE-L. The evaluation was executed on a public leaderboard[11], forked from EvalAI (Yadav et al., 2019), an open-source AI challenge hosting platform. In addition, 6 randomly selected summaries are selected from the top performing systems, to undergo human evaluation. The evaluation focuses on informativeness and readability.

4.2 Systems Overview

Nine systems participated in the task, with a total of 100 submissions. We will briefly describe eight of them, that submitted a research report describing their approach.

ARTU (El-Ebshihy et al., 2020) - The system generates an extractive summary which is based on the papers' abstract. Each sentence from the abstract becomes a query to an index that contains all papers' paragraphs. For each abstract sentence, a cluster that contains the top retrieved paragraphs is created. The final set of sentences is chosen based on the sentences LexRank value, their discourse (based on the section they belong to), and the size of the cluster.

AUTH (Gidiotis et al., 2020) - The authors propose an extractive summarization method that utilizes DANCER, a divide and conquer approach for long document summarization. DANCER (Gidiotis and Tsoumakas, 2020) helps to select key sections in the document to be summarized separately, for that each sentence in the article is classified to a section type. Then using PEGASUS based Transformer (Zhang et al., 2019) they are combined together to form an complete article summary.

CIST_BUPT (Li et al., 2020) - The system supports both an extractive and abstractive summaries using deep-learning architectures. For extractive summaries, they used RNN to compress and represent a sentence, and build a sentences relation graphs which are fed into the Graph Convolutional Network (GCN), and Graph Attention Network (GAN) to create a summary. For abstractive summaries, they used the gap-sentence method in (Zhang et al., 2015) to combine and transform all the data, and then T5 (Raffel et al., 2019), a transformer-liked pre-trained to fine-tune and generation.

GUIR (Sotudeh et al., 2020) - A summarization method that utilizes BERT summarizer (Liu and Lapata, 2019b). The idea is based on multi-task learning heuristic, in which two tasks are optimized. The first is a binary classification task, for sentence selection. The second is section prediction, in which the model predicts section labels associated with input sentences. The extractive network is then trained to optimize both tasks. The authors also propose an abstractive summarizer based on BART (Lewis et al., 2020) transformer that runs after the extractive summarizer.

IIITBH-IITP (Reddy et al., 2020) - The authors propose an extractive sentence classification method. They develop a deep learning architecture utilizing CNN to extract features, followed by MaxPooling and flattening for sentence representation and classification.

IITP-AI-NLP-ML (Mishra et al., 2020) - An unsupervised summarization technique that is used to extract salient sentences. First, article sentences are

[9] https://github.com/guyfe/LongSumm

[10] https://pypi.org/project/rouge-score/

[11] https://aieval.draco.res.ibm.com/challenge/39/

clustered together using various clustering methods (the authors considered various methods such as K-means (Lloyd, 1982) and DBScan (Ester et al., 1996)). Then, each cluster is ranked based on its centrality. Finally, salient sentences are selected from each cluster, taking into account cluster score, until the desired length of the summary.

Monash-Summ (Ju et al., 2020)- The system, inspired by SummPip (Zhao et al., 2020), proposes an unsupervised approach that leveraging linguistic knowledge to construct sentence graph. The graph nodes, which represent sentences, are further clustered. This enables the control of the summary length. Finally, for each cluster they considered the key phrases and discourse and created an abstractive sentence.

Summaformers (Roy et al., 2020) - To handle long documents, each section was allocated with a budget based on its contribution in the training data. Each section was summarized separately, using SummaRuNNer (Nallapati et al., 2017), a neural extractive summarizer.

4.3 Results

Table 4 reports the results of the 9 participating systems, 8 of them submitted a research report describing their system[12]. In order to compare between the systems we considered an average score of ROUGE-1, ROUGE-2, and ROUGE-L. Although some of the systems developed an abstractive variant, the highest ROUGE scores were obtained by leveraging extractive summarization techniques. The only system that reported abstrative summarization results, in the official leaderbaord, is *Monash-Summ*. Most of the systems except *ARTU* and *IITP-AI-NLP-ML* employ supervised learning approaches. The system that achieved the highest ROUGE average score is *GUIR*, with their multi-task learning heuristic. Second best is *Summaformers*, with about 3% lower ROUGE score.

In addition, we randomly selected 5 summaries from the top-3 ranked systems, namely: *GUIR*, *Summaformers* and *IITBH-IITP*, to be evaluated by experts. We asked them to rank the systems w.r.t coverage, and readability. For *coverage*, we asked to take into account how well the summary contains important, informative information conveyed in the text. For *Readability*, we asked to take into account fluency, coherence and grammatical correctness. From coverage perspective, all experts reported that *GUIR* summaries outperform the other systems, where the main issue with *Summaformers* and *IITBH-IITP* is that they mainly cover the introduction and related works sections. From readability perspective, the experts pointed out on several issues such as out of context formulas and reference to tables and figures, sentences are not sorted by the paper discourse, and footnotes that are clearly not relevant such as URLs, author's information, etc.

4.4 Discussion

Scientific documents can be characterized as long, structured, utilizing technical language (i.e., formulas, tables, definitions, etc.). Analyzing the summaries and reports of the participated systems shows that most of them considered the structure of the document while generating summaries, by utilizing sections and document discourse. From a language perspective, some systems utilized language models that were pre-trained on scientific corpora. However, we believe that more efforts should be focused on handling mathematical definitions, formulas, tables, and the text surrounding them. For example, it is not clear whether these entities should be treated differently than narrative text and whether they should be considered as atomic units that should not be compressed further.

Moreover, readability should play an important role in algorithmic design. Due to the nature of scientific documents and LongSumm length requirement, we believe this is even more challenging compared to traditional summarization tasks. This should have gotten more attention by the participating systems.

Finally, it was surprising to see that most evaluated systems are extractive and not abstractive. In the future we plan to extend this corpus, with the hope that LongSumm will help foster further research in this domain.

5 Conclusion

The First Scholarly Document Processing workshop (Chandrasekaran et al., 2020) comprise three summarization tasks, that each aimed to improve the state-of-the-art of scientific document summarization. In total, we received 18 submissions that addressed one or more of these tasks. It was a useful exercise to compare and contrast each of these summarization tasks, since they allowed re-

[12] Our analysis ignores *Wing* since they did not submit a system report as required

Table 4: ROUGE F-Measure and Recall evaluation on the official LongSumm test set. In addition, for each reported result, the Methodology columns indicate whether a reported result employs a **S**upervised or **U**nsupervised summarization technique.

System	**F-Measure**			**Recall**			**F-Measure average**	**Methodology**
	R-1	R-2	R-L	R-1	R-2	R-L		Supervised/ Unsupervised
GUIR	**53.11**	16.77	20.34	**54.60**	**17.28**	**20.90**	**30.07**	S
Wing	50.58	16.62	20.50	51.16	16.75	20.66	29.23	-
Summaformers	49.38	**16.86**	**21.38**	43.90	14.98	18.98	29.21	S
IIITBH-IITP	49.03	15.74	20.46	49.84	16.00	20.80	28.41	S
AUTH	50.11	15.37	19.59	46.93	14.23	18.18	28.36	S
CIST_BUPT	48.99	15.06	20.13	49.74	15.22	20.39	28.06	S
ARTU	48.03	14.76	18.04	46.78	14.28	17.43	26.94	U
IITP-AI-NLP-ML	46.46	14.61	19.58	47.43	14.86	19.95	26.88	U
Monash-Summ	49.16	12.80	18.31	49.35	12.76	18.33	26.76	S

searchers to explore their systems in different contexts, on different corpora, and for different audiences. Overall, what this efforts has shown is that the state of the art of summarizing scientific documents is neither in its nascency, nor a fully solved problem. We are interested in expanding task-based efforts in scholarly document summarization in future workshops, and investigating how scholarly documents differ or are similar to other texts. We are interested in collaborating with others in the NLP and AI-communities to investigate to what degree new technologies can be utilized and developed, to allow for a future where some of the work of tracking the scientific literature can be supported by machines. While CL-SciSumm has run for 6 editions and with the 2020 edition now set up two standard benchmark evaluation datasets for citation based summarization intended for use by researchers to aid in scientific discovery (breadth), LongSumm and LaySumm are inaugural tasks towards building systems that to improve understanding and dissemination of papers (depth).

Acknowledgements

CL-SciSumm would like to Microsoft Research Asia who funded the development of Cl-SciSumm corpus and the shared tasks from 2016 through 2018. We also thank Vasudeva Varma and colleagues at IIIT-Hyderabad, India and University of Hyderabad for their efforts in convening and organizing our annotation workshops in 2016-17. We acknowledge the advice of Min-Yen Kan, Hoa Dang, NIST, Lucy Vanderwende and Anita de Waard from the pilot stage of this task. We would also like to thank Rahul Jha and Dragomir Radev for sharing their software. We are grateful to Kevin B. Cohen and colleagues for their support, and for sharing their annotation schema and tools which have been indispensable for all six editions of CL-SciSumm.

The LongSumm task organizers would like to thank the blog authors Shagun Sodhani, Patrick Emami, Adrian Colyer, Alexander Jung, Joseph Paul Cohen, Hugo Larochelle, Elvis Saravia and to ShortScience.org who generously allowed them to share the content as part of the LongSumm dataset.

The LaySumm task organizers thank Darin McBeath at Elsevier who compiled the test and training data and Ilaria Meliconi, Virgina Prada Lopez and Victor Croes at Elsevier, who acted as domain experts for spot checking the results.

References

Dennis Aumiller, Satya Almasian, Philip Hausner, and Michael Gertz. 2020. UniHD@CL-SciSumm20: Citation Extraction as Search. In *SDP 2020*.

Maik Boltze, Anja Fischer, Artur Jurk Georg, and Keller Lorna Ulbrich. 2020. 1A-Team / Martin-Luther University Halle-Wittenberg@CL-SciSumm20. In *SDP 2020*.

Odellia Boni, Guy Feigenblat, Doron Cohen, Haggai Roitman, and David Konopnicki. 2020. A study of human summaries of scientific articles.

Ling Chai, Guizhen Fu, and Yuan Ni. 2020. NLP-PINGAN-TECH@CLSciSumm-20. In *SDP 2020*.

M. K. Chandrasekaran, G. Feigenblat, D. Freitag, T. Ghosal, Hovy. E., Mayr. P., M. Shmueli-Scheuer, and A De Waard. 2020. Overview of the first workshop on scholarly document processing (sdp). In *Proceedings of the First Workshop on Scholarly Document Processing (SDP 2020)*.

Muthu Kumar Chandrasekaran, Michihiro Yasunaga, Dragomir Radev, Dayne Freitag, and Min-Yen Kan.

2019. Overview and results: CL-scisumm shared task 2019. *arXiv preprint arXiv:1907.09854*.

Alaa El-Ebshihy, Annisa Maulida Ningtyas, Linda Andersson, Florina Piroi, and Andreas Rauber. 2020. ARTU / TU Wien and Artificial Researcher@ LongSumm 20. In *SDP 2020*.

Shai Erera, Michal Shmueli-Scheuer, Guy Feigenblat, Ora Peled Nakash, Odellia Boni, Haggai Roitman, Doron Cohen, Bar Weiner, Yosi Mass, Or Rivlin, Guy Lev, Achiya Jerbi, Jonathan Herzig, Yufang Hou, Charles Jochim, Martin Gleize, Francesca Bonin, Francesca Bonin, and David Konopnicki. 2019. A summarization system for scientific documents. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, page 226–231. AAAI Press.

Alexios Gidiotis, Stefanos Dimitrios Stefanidis, and Grigorios Tsoumakas. 2020. AUTH@CL-SciSumm20, CL-LaySumm20, LongSumm20. In *SDP 2020*.

Alexios Gidiotis and Grigorios Tsoumakas. 2020. A divide-and-conquer approach to the summarization of academic articles. *arXiv preprint arXiv:2004.06190*.

Rong Huang and Kseniia Krylova. 2020. Team MLU@CL-SciSumm20: Methods for Computational Linguistics Scientific Citation Linkage. In *SDP 2020*.

Kokil Jaidka, Muthu Kumar Chandrasekaran, Devanshu Jain, and Min-Yen Kan. 2017. The cl-scisumm shared task 2017: Results and key insights. In *BIRNDL@ SIGIR (2)*, volume 2002, pages 1–15. CEUR.

Kokil Jaidka, Muthu Kumar Chandrasekaran, Sajal Rustagi, and Min-Yen Kan. 2018. Insights from cl-scisumm 2016: the faceted scientific document summarization shared task. *International Journal on Digital Libraries*, 19(2-3):163–171.

Kokil Jaidka, Michihiro Yasunaga, Muthu Kumar Chandrasekaran, Dragomir Radev, and Min-Yen Kan. 2019. The cl-scisumm shared task 2018: Results and key insights. *arXiv preprint arXiv:1909.00764*.

Jiaxin Ju, Ming Liu, Longxiang Gao, and Shirui Pan. 2020. Monash-Summ@LongSumm 20 SciSummPip: An Unsupervised Scientific Paper Summarization Pipeline. In *SDP 2020*.

Seungwon Kim. 2020. Using Pre-Trained Transformer for a better Lay Summarization. In *SDP 2020*.

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. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2125–2131.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pretraining for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Lei Li, Yang Xie, Wei Liu, Yinan Liu, Yafei Jiang, Siya Qi, and Xingyuan Li. 2020. CIST@CLSciSumm-20, LongSumm 2020: Automatic Scientific Document Summarization. In *SDP 2020*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*, volume 8. Barcelona, Spain.

Yang Liu and Mirella Lapata. 2019a. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3730–3740, Hong Kong, China. Association for Computational Linguistics.

Yang Liu and Mirella Lapata. 2019b. Text summarization with pretrained encoders. *arXiv preprint arXiv:1908.08345*.

S. P. Lloyd. 1982. Least squares quantization in pcm. *IEEE Trans. Inf. Theory*, 28:129–136.

Santosh Kumar Mishra, Kundarapu Harshavardhan, Naveen Saini, Sriparna Saha, and Pushpak Bhattacharyya. 2020. IITP-AI-NLP-ML@CLSciSumm-20, CL-LaySumm 2020, LongSumm 2020. In *SDP 2020*.

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 Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 3075–3081. AAAI Press.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer.

Saichethan Miriyala Reddy, Naveen Sainiand Naveen Saini, Sriparna Saha, and Pushpak Bhattacharyya. 2020. IIITBH-IITP@CL-SciSumm20, CL-LaySumm20, LongSumm20. In *SDP 2020*.

Eleanor H. Rosch. 1973. Natural categories. *Cognitive Psychology*, 4(3):328 – 350.

Sayar Ghosh Roy, Nikhil Pinnaparaju, Risubh Jain, Manish Gupta, and Vasudeva Varma. 2020. Information Retrieval and Extraction Lab, IIIT-H @ LaySumm 20, LongSumm 20. In *SDP 2020*.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Eva Sharma, Chen Li, and Lu Wang. 2019. Bigpatent: A large-scale dataset for abstractive and coherent summarization. *arXiv preprint arXiv:1906.03741*.

Sajad Sotudeh, Arman Cohan, and Nazli Goharian. 2020. GUIR @ LongSumm 2020: Learning to Generate Long Summaries from Scientific Documents. In *SDP 2020*.

Wenliang Dai Tiezheng Yu, Dan Su and Pascale Fung. 2020. Dimsum @LaySumm 20. In *SDP 2020*.

Anjana Umapathy, Karthik Radhakrishnan, Kinjal Jain, and Rahul Singh. 2020. CiteQA@CL-SciSumm20. In *SDP 2020*.

Deshraj Yadav, Rishabh Jain, Harsh Agrawal, Prithvijit Chattopadhyay, Taranjeet Singh, Akash Jain, Shiv Baran Singh, Stefan Lee, and Dhruv Batra. 2019. Evalai: Towards better evaluation systems for ai agents.

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.

Heng Zhang, Lifan Liu, Ruping Wang, Shaohu Hu, Shutain Ma, and Chengzhi Zhang. 2020. IR&TM-NJUST @ CLSciSumm-20. In *SDP 2020*.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J Liu. 2019. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. *arXiv preprint arXiv:1912.08777*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15.

Jinming Zhao, Ming Liu, Longxiang Gao, Yuan Jin, Lan Du, He Zhao, He Zhang, and Gholamreza Haffari. 2020. Summpip: Unsupervised multi-document summarization with sentence graph compression. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 1949–1952.