IIITBH-IITP@CL-SciSumm20, CL-LaySumm20, LongSumm20

Saichethan Miriyala Reddy*, **Naveen Saini**†, **Sriparna Saha**†, **Pushpak Bhattacharyya**†
*Indian Institute of Information Technology, Bhagalpur
†Indian Institute of Technology, Patna
miriyala.cse.1725@iiitbh.ac.in, {naveen.pcs16, sriparna, pb}@iitp.ac.in

Abstract

In this paper, we present the IIIT Bhagalpur and IIT Patna team's effort to solve the three shared tasks namely, CL-SciSumm 2020, CL-LaySumm 2020, LongSumm 2020 at SDP 2020. The themes of these tasks are to generate medium-scale, lay and long summaries, respectively, for scientific articles. For the first two tasks, unsupervised systems are developed, while for the third one, we have developed a supervised system. The performances of all the systems are evaluated on the associated datasets with the shared tasks in term of well-known ROUGE metric.

1 Introduction

Due to a lot of research going into the computational linguistic (CL) domain as well as in other domains, the rate of publishing scientific articles has been increased and will continue to expand (Nallapati et al., 2017, 2016; Jaidka et al., 2019). This makes the researchers challenging to update them with the up-to-date advancements. A survey (review) article may help the researcher to have a gist of the recent advancements. But, writing a survey paper is a very laborious and time-consuming task. This challenge demands summarization of scientific articles (Cohan and Goharian, 2018; Conroy and Davis, 2018) by providing their summary in a few words and then prepare the survey article .

But sometimes, for niche practitioners, the published and survey articles may be difficult to understand. To make them relevant for the non-practitioners and to benefit all the researchers, it is indeed a need to outline the contribution of research articles in lay language.

The current paper demonstrates the participation of IIIT Bhagapur and IIT Patna team in three shared tasks namely, *CL-SciSumm 2020, LongSumm 2020 and CL-LaySumm 2020*, at first workshop on Scholary Document Processing[1], 2020 (Chandrasekaran et al., 2020). The theme of these tasks is to generate medium-scale, long and Lay summaries, respectively. Here, Lay summary means a textual summary which is intended for non-technical audience. The scientific articles used for the first and third tasks are related to computational linguistic domain. While, for the second task, scientific articles cover distinct domains: archeology, epilepsy, and materials engineering. In current paper, all these tasks are posed as extractive summarization (Saini et al., 2019) problems where a subset of sentences are selected from scientific articles based on their relevance. For CL-LaySumm and CL-SciSumm, we have developed the system based on the maximal marginal relevance (MMR) (Carbinell and Goldstein, 2017) which considers novelty and informativeness of sentences with respect to what is already included in the summary. And, for LongSumm, our system utilizes neural network based approach. More descriptions about these tasks including datasets and methodology used, are provided in the subsequent sections. The performances of the systems are evaluated in terms of ROUGE (1-gram, 2-gram, and L) metrics on the provided dataset.

2 CL-SciSumm 2020

CL-SciSumm 2020 is the sixth Computational Linguistics Scientific Document Summarization Shared Task which aims to generate summaries of scientific articles not exceeding 250 words. The associated dataset for the task is provided with a Reference Paper (RP) (the paper to be summarized) and 10 or more citing Papers (CPs) containing citations to the RP, which are used to summarise RP. It includes two more sub-tasks: (a) *Task 1(A)*- iden-

[1] https://ornlcda.github.io/SDProc/index.html

Proceedings of the First Workshop on Scholarly Document Processing, pages 242–250

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

tifying the text-spans in the reference article that mostly reflect the citation contexts (i.e., citances that cite the RP) of the citing articles; (b) *Task 1(B)*- categorizing the identified text-spans into a predefined set of facets. Generation of structured summary for scientific document summmarization using the identified text-spans is covered in *Task 2*.

2.1 Dataset Description

The dataset associated with CL-SciSumm 2020 shared task, consists of 40 annotated scientific articles and their citations for training. In addition to this, a corpus of 1000 documents released as a part of ScicummNet (Yasunaga et al., 2019) dataset for scientific document summarization is readily available for training. For testing, a blind test set of 20 articles used for CL-SciSumm 2018 (Jaidka et al., 2019) and 2019 (Chandrasekaran et al., 2019) shared tasks, is again used for the current shared task.

2.2 Methodology

In this section, we have discussed the system developed for Task 1 and Task 2. The corresponding flowchart is shown in Figure 1.

2.2.1 Task 1(A)

For a given reference paper (RP), in order to identify the reference text-spans using citation context, we have used an unsupervised approach where we have extracted the top 5 sentences by calculating cosine similarity between each citance and sentences of the RP. These 5 sentences are considered as cited/reference text spans. Note that before calculating the similarity, we have converted the text-space into a (numeric) vector-space for which we have utilized different types of sentence embeddings namely, Albert (Beltagy et al., 2019a), ELMO (Peters et al., 2018), fastText (Athiwaratkun et al., 2018), SciBERT (Beltagy et al., 2019a), Universal Sentence Encoder (Cer et al., 2018), XLNET (Yang et al., 2019), which are capable of capturing the semantics of the sentences. Thus, in total, six systems are developed for Task 1(A).

2.2.2 Task 1(B)

For identifying discourse facets (Hypothesis, Implication, Aim, Results and Method) of cited text spans, we have used a voting based method. A supervised multi-class classification model, based on the Gradient Boosting (La Quatra et al., 2019; Li et al., 2008), is trained in order to assign a facet to each cited text span. Training data statistics are described in Table 1. In our approach, we have extracted top 5 text spans for each citance in Task 1(A). We have used our trained model to identify facet for each cited text span. Later we have used a voting method to finalize facet for each citance.

Section	no. of sentences
Aim	78
Method	823
Hypothesis	23
Implication	91
Results	121
Total	1136

Table 1: Task 1B data statistics

2.2.3 Task 2

For generating structured summary of 250 words, we have used the unique sentences extracted in Task 1(A) (i.e., cited text spans) as the candidate set of sentences. This approach is known as citation-based summarization. For this purpose, a diversity-based unsupervised measure namely, maximal marginal relevance (MMR), inspired from (Carbinell and Goldstein, 2017) which is a linear combination of informativeness (with respect to documents consisting of chosen candidate sentences) and novelty of the sentence (with respect to sentences already included in the summary) is utilized. Mathematically, it is expressed as

$$MMR_1 = \lambda_1 Sim_1(Q, D) - (1 - \lambda_1) Sim_2(Q, d) \quad (1)$$

where, Q is the current sentence, D is the list of extracted sentences in Task 1(A), d is the generated summary till that point of time, Sim_1 is the similarity of a sentence with respect to all other sentences in the document, Sim_2 is the similarity of current sentence with the sentences that are already included in the summary. Note that for representation of sentences into vector form, we have used CountVetorizer[2] which counts the term-frequency of each term in the article.

The authors of the paper (Jaidka et al., 2017) which was on summarizing scientific articles mentioned that system performance using ROUGE mea-

[2] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html

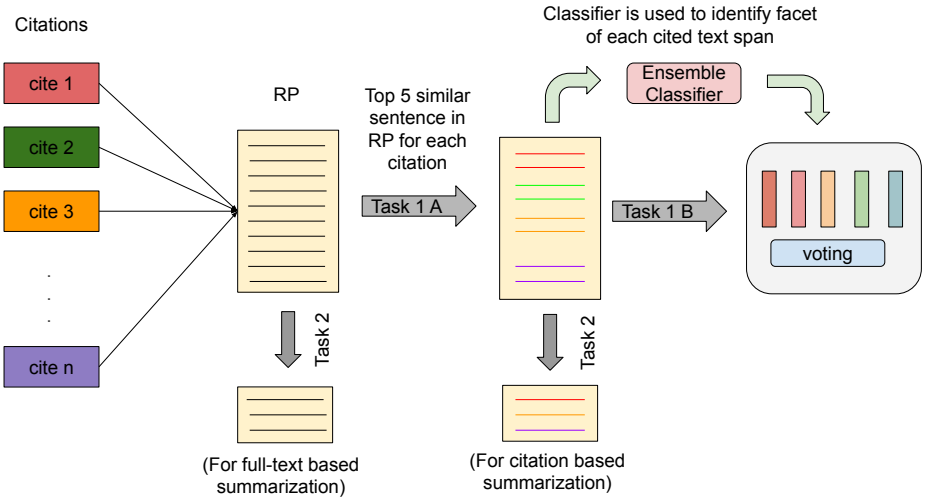

Figure 1: Proposed Architecture for Task 1(A) and Task 1(B) for CL-SciSumm 2020.

sure is not always lenient than sentence overlap F1 scores. They demonstrated how the ROUGE score is biased to prefer shorter sentences over longer ones. Motivated by this, we have proposed a variant of MMR by incorporating length of the sentence and is expressed as

$$MMR_2 = MMR_1 - \frac{\lambda_2}{L} \quad (2)$$

where, L is the length of the current sentence.

Total twelve systems are developed using the citation-based approach in 6 different semantic spaces (refer to Section 2.2.1), each utilizing MMR_1 and MMR_2 for summary generation. To show the potentiality of citation-based summarization, we have also developed full-text based summarization where we have considered total sentences available in the scientific article as the candidate set of sentences for summary generation and utilized the MMR_2 for summary generation. Thus, in total, 13 systems are submitted in the CL-SciSumm 2020 shared task.

2.3 Discussion of Results

We have submitted a total of 13 system runs out of which 6 runs are for both Task 1(A) and Task 1(B) utilizing different semantic space. Rest of the 7 runs are only for the Task 2. Results obtained by our different runs for Task 1(A) and Task 1(B) are illustrated in Table 2 and Table 3, respectively. For Task 2, we have generated a single summary for each reference paper using MMR_1 and MMR_2 for different embeddings. Out of 13 system runs, 12 are citation-based, and remaining one is full text based. Results obtained are illustrated in Table 4. For task 1A, & 1B, the best results are obtained using universal sentence encoder for sentence embedding. For Task 2, our enhanced diversity based sentence selection approach, i.e., MMR_2, has performed better than existing maximum marginal relevance model (MMR_1). It is important to note that MMR_2 is tested with different embedding space; but all gives the similar results. Therefore, in Table 4, we have mentioned only MMR_2 as representive of those runs. From Table 4, we can also infer that citation based summarization has better sentence overlaps compared to full text based summarization (last row of Table 4). Note: Using MMR_2 we have obtained exact same results irrespective of the embeddings used. So in Table 4, we have added a single entry MMR_2 representative of those 6 systems.

Poor performance of our system for abstract summary can be explained since our approach tries to focus more on coverage, and diversity. Abstract of any scientific article lies in the starting part and since we are not considering position in our pro-

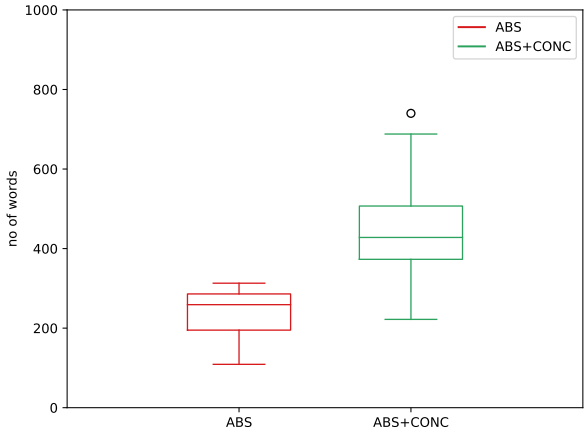

Figure 2: LaySumm test data statistics.

posed approach thus lesser sentence overlaps.

3 CL-LaySumm 2020

The CL-LaySumm 2020, which is the first shared task[3] for Lay summary generation, is for automatic generation of Lay summary in 70-100 words which is readable and easily understandable by the general pubic. In other words, given a full-text paper and its abstract, the task is to generate a Lay Summary of the specified length of that paper.

3.1 Description of Dataset

The dataset is provided with 600 scientific articles with its, abstract, full-text and corresponding lay summary (gold summary) of around 70-100 words. The test data consists of 37 articles (out of 600 articles). Test data statistics in terms of number of words are shown in Figure 2.

3.2 Methodology

In this section, we have discussed the methodology used for Lay summary generation. Similar to CL-SciSumm, we have considered this problem as a sentence selection problem where relevant sentences are selected from the document to generate the summary.

Similar to Cl-SciSumm, here also, we have used both variants of maximum marginal relevance (MMR) mentioned in Eq 1 and Eq 2 for generating summary. As abstract (Let us call it as ABS) conveys the outline of the paper; therefore, we have compared the summary generated using different variants of MMR with the ABS. Other comparisons are done with the original Lay summary when using (a) the full-text of the article; (b) abstract (ABS) and conclusion (CON) of the paper.

Note that goal of generating lay summary is to create a human readable summary for non-technical audience. To avoid scientific jargon in the generated summary, we have proposed a three step process (let us call it as **CWR**: Complex Word Removal) where firstly we identify complex words from given sentence, then generate similar words of identified complex words, replace with most suitable word from the generated list. In this paper, we have only identified complex words and removed them, pseudo code for identifying complex words is given in Algorithm 1.

Algorithm 1: CWR

```
Result: List of Complex Words
set W = set of unique words from generated
  summary;
set KB = list of words in glove or wordnet;
set len = len(W);
initialize CWR = [];
/*list of complex words*/;
for i ← 0 to len do
    word = W[i];
    cleanWord = clean(word);
    /*remove unwanted symbols*/;
    lemWord = lemmatisation(cleanWord);
    if lemWord not in KB then
        CWR.append(word)
    end
end
```

3.3 Discussion of Results

Results obtained using MMR_1 and, MMR_2 on ABS, FULL-TEXT and ABS+CON, are reported in Table 5. From this Table, it can be observed that by considering length in to the MMR_1, i.e., Eq. (2) and generating summary using the abstract of the article helps in improving the performance of the system in comparison to MMR_1. We have also illustrated how ROUGE-1 F score varied with λ_1 and, λ_2 in Table 6. Note that these parameters play important roles in generating the informative and novel Lay summary and are the parts of Eqs. (1) and (2). The best values of the parameters used in MMR_1 and MMR_2 are highlighted (in bold) in Table 6, i.e., for MMR_1, $\lambda_1 = 0.75$ and for MMR_2, $\lambda_1 = 0.75$, $\lambda_2 = 0.20$, are the best val-

[3] https://ornlcda.github.io/SDProc/sharedtasks.html#laysumm

Variant	Task 1A					
	Precision		Recall		F1	
	Micro	Macro	Micro	Macro	Micro	Macro
ALBERT	0.0202	0.0194	0.0953	0.0937	0.0333	0.0322
ELMO	0.0126	0.0131	0.0594	0.0622	0.0208	0.0216
FastText	0.0357	0.0362	0.1685	0.1727	0.059	0.0598
SciBERT	0.0114	0.0106	0.0539	0.0502	0.0188	0.0175
USE	**0.0469**	**0.0471**	**0.221**	**0.2246**	**0.0773**	**0.0779**
XLNET	0.0029	0.0032	0.0138	0.0146	0.0048	0.0053

Table 2: Performance of different system runs for Task 1A

Variant	Task 1B					
	Precision		Recall		F1	
	Micro	Macro	Micro	Macro	Micro	Macro
ALBERT	0.4649	0.4102	0.0789	0.0716	0.1349	0.122
ELMO	0.3333	0.2922	0.0461	0.0463	0.0809	0.0799
FastText	0.3882	0.4386	0.0985	0.0991	0.1571	0.1617
SciBERT	0.2644	0.2715	0.0341	0.0311	0.0604	0.0558
USE	**0.4900**	**0.4849**	**0.1469**	**0.1461**	**0.2261**	**0.2245**
XLNET	0.0517	0.0403	0.0044	0.0049	0.0082	0.0088

Table 3: Performance of different system runs for Task 1B

Variant	ABSTRACT		COMMUNITY		HUMAN	
	R-2	R-SU4	R-2	R-SU4	R-2	R-SU4
ALBERT + MMR_1	0.06548	0.01006	0.24342	0.12235	0.09604	0.01873
ELMO + MMR_1	0.09119	0.01435	0.2328	**0.14525**	0.11379	0.02512
FastText + MMR_1	0.08718	0.01111	0.25724	0.12458	0.10957	0.01945
SciBERT + MMR_1	0.13277	0.01211	0.18978	0.07994	0.14022	0.01846
USE + MMR_1	0.10521	0.01438	**0.27462**	0.13962	0.12955	0.02507
XLNET + MMR_1	0.05816	0.00825	0.17212	0.09749	0.08559	0.0176
MMR_2	**0.15067**	**0.07851**	0.13976	0.07268	**0.15073**	**0.10237**
MMR_2 (full text)	0.03909	0.03708	0.12305	0.06701	0.05206	0.0503

Table 4: Performance (F1 scores) of different system runs for Task 2

ues. Here, λ_1 represents the diversity factor as we increase λ_1, diversity of generated summary decreases. Reader may have in mind that we are using high value of λ_1, i.e., 0.75 and thus, the summary may have less coverage. Since average number of words in ABS are around 250 (Figure 2) and our task is to find lay summary around 100 words; therefore, we have used higher λ_1. Whereas λ_2 tries to maximize Rouge score. As in Table 5, abstract is shown to be good for the summary generation; therefore, we have executed the Algorithm 1 using the same (i.e., ABS). The results attained by CWR using different variants of MMR are shown in Table 7. After observing the results, it is clear that there is not much difference among-st the best results of Table 5 and 7, or in other words, the results of Table 7 are quite less than those reported in Table 5.

Error analysis of CWR: Some of the common issues associated with identifying complex words are finding lemma. Lemmatization usually refers to performing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma. Few common scientific terms which are not present in lexical databases like wordnet (Miller, 1995) can be important and trivial in the context of the paper. For example words like ”hepatocellular”, ”carcinoma”

Variant	Data	F1 Scores		
		R-1	R-2	R-L
MMR_1	ABS	0.4009	0.1679	0.2239
MMR_2	ABS	**0.4048**	**0.1690**	**0.2244**
MMR_1	ABS+CON	0.3837	0.1411	0.2050
MMR_2	ABS+CON	0.3855	0.1394	0.2055
MMR_1	FULL	0.2835	0.0604	0.1609
MMR_2	FULL	0.2875	0.0628	0.1592

Table 5: Results attained using MMR and it's variant for CL-LaySumm 2020. Here, R in second row stands for 'ROUGE'.

λ_1	λ_2	ROUGE 1-F
0.25	0.0	0.3876
0.50	0.0	0.3940
0.75	0.0	**0.4009**
1.00	0.0	0.3971
0.75	0.1	0.4031
0.75	0.2	**0.4048**

Table 6: Study of parameters used in MMR_1 and MMR_2 for Lay Summary generation. Here, we have used only ABSTRACT for generating summary.

Variant	Data	F1 Scores		
		R-1	R-2	R-L
$CWR + MMR_1$	ABS	0.3986	0.1586	0.2187
$CWR + MMR_2$	ABS	**0.4033**	**0.1614**	**0.2209**

Table 7: Results attained by applying CWR on the generated summary using abstract (ABS). Here, R in second row stands for 'ROUGE'.

etc., are trivial for paper (S016882782030009X) but are not present in wordnet vocabulary. Therefore, our result using CWR (Table 7) underperform that by MMR_2 (Table 5), thus demanding more sophisticated model.

4 LongSumm 2020

Most of the existing works on scientific document summarization focus on generating a summary of shorter length (maximum up to 250 words). Such type of length constraint can be sufficient when summarizing news articles, but for scientific articles, the summary requires expertise in the scientific domain to understand it. LongSumm 2020 shared task addresses this issue by generating longer summaries (up to 600 words) of scientific articles.

4.1 Dataset Description

The training corpus for this task includes 1705 extractive summaries, and 531 abstractive summaries of NLP/ML scientific papers. The extractive summaries are based on video talks from associated conferences (Lev et al., 2019), while the abstractive summaries are from blog posts created by NLP and ML researchers. The test set consists of 22 research papers for both extactive and abstractive summarization and task is to generate a summary of 600 words. In the current paper, we have focused only on the extractive summarization of LongSumm.

4.2 Methodology

To solve the LongSumm in an extactive way, we have utilized the neural network based approach, i.e., convolution neural network (Kim, 2014). The sentences which are part of the summary are assigned 1 and remaining sentences are assigned 0. In other words, we have posed this task as a binary classification problem where task is to identify whether the given sentence can be a part of the summary or not. Positional embedding is also used along with sentence embedding. The detailed methodology used in our CNN is described below:

1. **Convolution:** Authors of (Kim, 2014) showed that CNN with one layer of convolution performs remarkably well for sentence classification tasks. Therefore, we have used one dimensional CNN for extracting features from sentences as described below mathematically:

$$c_i = g(W_f^T \dot{X}_{i:i+m-1} + b) \quad (3)$$

$$c = [c_1, c_2, c_3, ..., c_{n-m+1}] \quad (4)$$

where b is the bias term, g is a non-linear activation function, W_f, m and X are convolution filter, window size and concatenation vector, respectively.

2. **MaxPooling:** Pooling is a down sampling operation. In max pooling, each pooling operation selects the maximum value of the current view and thus reduces the size yet preserves features as shown below:

$$h_l = max(c) \quad (5)$$

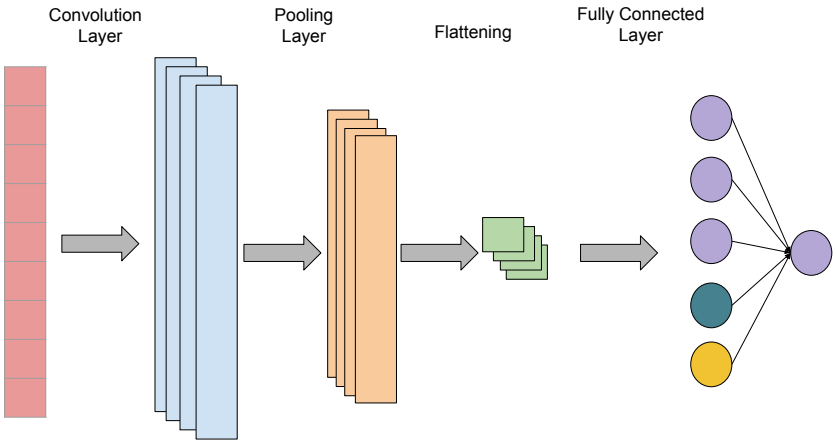

Figure 3: Architecture used for LongSumm2020.

$$h = [h_1, h_2..., h_k]^T \tag{6}$$

where h is hidden representation of sentence after convolution.

3. **Positional Embedding:**

   In any document, regardless of the domain, more relevant sentences can be found in some sections of the document like the leading paragraph of the document (Saini et al., 2019). Particularly scientific articles are structured in a way that sentences at start (abstract) are more informative as represented below.

$$p_i = \frac{1}{1+i} \tag{7}$$

   where p_i is the i^{th} $(0 \leq i < N)$ sentence in the article. Higher the score for a sentence, more informative it is. Therefore, positional embedding is also utilized in our CNN framework.

4. **Flattening:** After the max-pooling layer, we obtain the penultimate layer h (Eq. 6), which is the vector representation of the input sentence obtained from CNN. We have also fed sentence position encoding (h_p) as additional feature.

$$h^* = [h, h_p] \tag{8}$$

   where h^* is the semantic representation obtained from CNN and h_p is position encoding represented as

$$h_p = [p_1, p_2..., p_k]^T \tag{9}$$

   To avoid overfitting, we have used regularization as mentioned in Eq 10.

$$\hat{y} = \sigma(w_r(h^* \otimes r) + b_r) \tag{10}$$

   And finally, we have used sigmoid function as per Eq 11 for obtaining probability scores:

$$\sigma(\hat{y}) = \frac{1}{1+e^{-\hat{y}}} \tag{11}$$

Note that we have considered sigmoid probability for assigning ranks to sentences and used those for sentence selection to be included in the summary till the length constraint is satisfied.

4.3 Experimental setup

For our experimentation, we have used SciBert (Beltagy et al., 2019b) to get the sentence embeddings as it is trained on a large multi-domain corpus

of scientific publications to improve performance on many scientific NLP tasks like summarization (Gabriel et al., 2019) and relation extraction (Sung et al., 2019). For convolution layer, we have used 600 filters, and 3 kernels with ReLU as our activation function. For Pooling, we have used pool size of 2. We train the model for 10 epochs with the Adadelta optimizer.

4.4 Discussion of Results

We have submitted 4 systems for LongSumm shared task. Out of 4, two systems are based on CNN architecture. The key difference between two neural models is essentially the limit of the number of words for summary generation, i.e., the first system (CNN_1) uses a strict 600 words and the second system (CNN_2) maintains on an average of 600 words for generating summaries. For other two systems, we have used MMR_1 and MMR_2 using same hyper parameters as LaySumm (Section 3.3). The results obtained for LongSumm 2020 task are reported in Table 8. From this Table it can be inferred that (a) CNN_2 performs better in term of Rouge-2 and Rouge-L F1-measure, but in terms of Rouge-1 F1-measure, MMR_2 performs the best. Training vs. testing accuracy for the results obtained using CNN_2 are shown in Figure 4.

Model	Rouge 1-F	Rouge 2-F	Rouge L-F
MMR_1	0.4958	0.1415	0.1815
MMR_2	**0.4960**	0.1418	0.1872
CNN_1	0.4840	0.1535	0.1993
CNN_2	0.4903	**0.1574**	**0.2046**

Table 8: Results of our top system runs for LongSumm 2020 shared task

5 Conclusion and Future work

We have investigated the effects of using maximal marginal relevance (MMR) in developing the systems for three shared tasks: CL-SciSumm, CL-LaySumm, and LongSumm 2020. Another variant of MMR is also proposed by incorporating a length-based feature. For LongSumm, we have also investigated the effect of using a convolution neural network. As the goal of LaySumm is to generate Lay summary, which is understandable for a non-technical audience, we have tried a common word removal approach using the lexical database like WordNet, which fails due to non-presence of scientific terms. In the future, we would like to develop a more sophisticated approach for LaySumm generation.

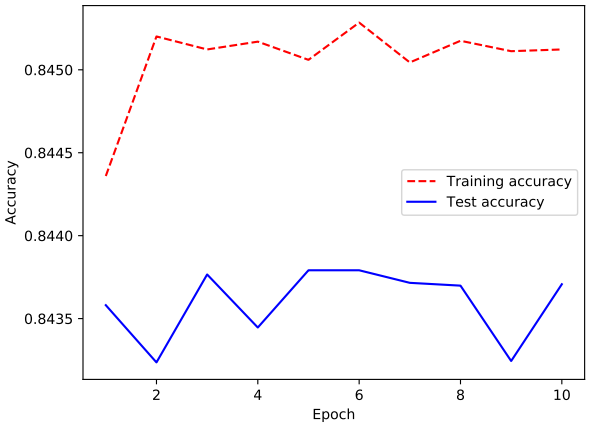

Figure 4: Training vs validation accuracy

References

Ben Athiwaratkun, Andrew Gordon Wilson, and Anima Anandkumar. 2018. Probabilistic fasttext for multi-sense word embeddings. *arXiv preprint arXiv:1806.02901*.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019a. Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019b. Scibert: A pretrained language model for scientific text. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 3613–3618. Association for Computational Linguistics.

Jaime Carbinell and Jade Goldstein. 2017. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *ACM SIGIR Forum*, volume 51, pages 209–210. ACM New York, NY, USA.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

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)*.

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*.

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.

John M Conroy and Sashka T Davis. 2018. Section mixture models for scientific document summarization. *International Journal on Digital Libraries*, 19(2-3):305–322.

Saadia Gabriel, Antoine Bosselut, Ari Holtzman, Kyle Lo, Asli Çelikyilmaz, and Yejin Choi. 2019. Cooperative generator-discriminator networks for abstractive summarization with narrative flow. *ArXiv*, abs/1907.01272.

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*.

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*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Moreno La Quatra, Luca Cagliero, and Elena Baralis. 2019. Poli2sum@ cl-scisumm-19: Identify, classify, and summarize cited text spans by means of ensembles of supervised models. In *BIRNDL@ SIGIR*, pages 233–246.

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. *arXiv preprint arXiv:1906.01351*.

Ping Li, Qiang Wu, and Christopher J Burges. 2008. Mcrank: Learning to rank using multiple classification and gradient boosting. In *Advances in neural information processing systems*, pages 897–904.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

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

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

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):e0223477.

Chul Sung, Tejas I. Dhamecha, Swarnadeep Saha, Tengfei Ma, V. Pulla Reddy, and Rishi Arora. 2019. Pre-training bert on domain resources for short answer grading. In *EMNLP/IJCNLP*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763.

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.