Exploring Summarization of Scientific Tables: Analysing Data Preparation and Extractive to Abstractive Summary Generation

Monalisa Dey*
Jadavpur University, Kolkata, INDIA
Sainik Kumar Mahata†
Institute of Engineering and Management, Kolkata, INDIA
Dipankar Das‡
Jadavpur University, Kolkata, INDIA

Abstract

This research paper focuses on the challenges involved in extracting relevant information from the vast amount of textual data available in digital resources. Specifically, we address the complexities associated with tables in scientific papers, which contain crucial data but can be difficult to understand and summarize. In fact, the data required for training such systems is scarce. To overcome this challenge, we propose the development of a high-quality corpus consisting of summaries, both extractive and abstractive, from table content. To develop such a corpus containing pertinent extractive-abstractive summary pair, we used two approaches, namely, rule-based and template-based. The developed dataset was then validated using various automated and manual metrics. Subsequently, two models, viz., T5 and Seq-to-Seq, were trained on this dataset, to generate abstractive summaries from extractive ones. By using extractive summaries as a starting point, the system can produce new sentences that capture the essence of the tables rather than all the specifics. We have used standard evaluation measures like BLEU, ROUGE-L, Adequacy, and Fluency to evaluate both systems. BLEU and ROUGE-L scores of 58.2 and 0.31 for fine-tuned T5 model and 16.25 and 0.18 for the Seq-to-Seq model suggest that our developed dataset can produce coherent abstractive summaries.

Key Words: Summarization; extractive; abstractive; rule-based; evaluation metrics; template-based; T5; Seq-to-Seq

1 Introduction

In this digital age, cloud resources like websites, social media platforms, news outlets, and communication channels, are seeing an unprecedented increase in information. A huge amount of textual data, generated from different places like academic books, research papers, legal proceedings, health sector, etc., presents a huge challenge when you need to have access to relevant data [1]. Due to this, the development of automatic techniques for extraction as well as summarization of textual resources has assumed significant importance in our lives. Automatic text summarization can be categorized into two types; extractive and abstractive. The extractive approach consists of selecting and extracting crucial sentences/paragraphs from the original document, to create a coherent summary. In contrast, abstractive summarization aims to develop a summary that incorporates novel lexemes and phrases.

In our work, we primarily concentrate on scientific papers. Multiple research papers and articles inundating our lives pose difficulties in swiftly and effectively extracting pertinent information from non-textual elements such as tables, graphs, figures, and flowcharts [2]. Tables, an integral part of scientific papers, provide an efficient medium to present intricate information concisely and in an organized manner. Nevertheless, tables present a challenge to conventional retrieval methods due to the fusion of content and presentation they embody. Therefore, the provision of a succinct summary of table data assumes a paramount role. This concise summary grants researchers the ability to comprehend the information without undertaking the task of reading the entire paper, thereby saving valuable time and effort. By offering a bird's-eye view of the data, users can swiftly grasp the essence of the table content without delving into complexities, empowering them to go through more results.

Table content summarization systems are therefore needed to extract table summaries effectively. However, the development of such systems faces a major challenge: the lack of appropriate corpora for training and evaluating summarization algorithms. A dataset suitable for summarizing patient information may prove futile when attempting to summarize weather reports or cricket scores.

To address the challenge posed by the scarcity of suitable training and testing corpus for scientific table summarization systems, we decided to develop a comprehensive corpus containing table-content summaries. Our study aims to generate two distinct types of summaries, extractive and abstractive. Our first contribution is constructing the table-content summary corpus. To do this we employed multiple techniques, encompassing information extraction, and extensive validation. Our approach entailed the creation of two types of summaries: extractive and abstractive. In extractive summaries, we carefully selected pertinent phrases from the text of the paper that conveyed information about the table,

*Email: monalisa.dey.21@gmail.com
†Email: sainik.mahata@gmail.com
‡Email: dipankar.dipnil2005@gmail.com

while in abstractive summaries, we extracted the accompanying captions for each table. These methodologies collectively furnished us with a robust and comprehensive corpus of table-content summaries, tailor-made for training and testing table summarization systems.

Our next contribution is to develop methods for selecting pertinent extractive-abstractive summary pair. To accomplish this, we used two approaches, namely, rule-based and template-based. The most crucial task was to recognize and separate the captions from the body content for each table in the rule-based strategy. Again, correctly extracting table mentions and citations from the body text was also very challenging. To overcome these obstacles, methods that could precisely parse the document and extract the relevant data had to be developed.

In the template-based approach, we also developed template summaries using several techniques like TF-IDF and Transition based approaches. These techniques identify a set of terms or a template, that is significant in representing the summary of that table. Both automatic and manual evaluation techniques are utilized for evaluating the quality of this dataset.

Finally, we compared both methods and chose the better-performing method for our next tasks. We have used the most famous ROUGE [16] and BLEU [25] tools to evaluate our results.

Subsequently, after solving every challenge, we successfully developed the corpus, which will serve as an important resource for any researcher interested in table summarization.

The next contribution in this paper is the development of an extractive-to-abstractive summary generation system, which would take as input the chosen extractive summary for each table and would generate an abstract representation of the same. For this, we employed two models; a transformer-based T5 model [1] that was fine-tuned using our own data as well as a seq-to-seq model. Figure 1 shows the flow diagram for the whole process.

The rest of the paper is structured as follows. The most recent developments in this area of research are covered in Section 2. The procedure for developing the dataset is described in Section 3. The methods and models for extractive to abstractive summary creation are described in Section 4, and the conclusion is presented in Section 5.

2 Related Work

The effort to summarize a huge body of information has given rise to two main techniques: extractive and abstractive summarization. Relevant sentences are carefully selected from the source document and presented together in extractive summarization. Abstractive methods, however, follow a different route. Here, new phrases are developed. This section includes a review of literature on extractive to abstractive summarizing, as well as table-based summarization that highlights significant contributions to the field of study.

[1]https://huggingface.co/docs/transformers/model_doc/t5

Nallapati et al. [24] introduced an abstractive text summarization approach using seq-to-seq RNNs and explored techniques to improve the quality of generated summaries. Gehrmann et al. [8] proposed a bottom-up abstractive summarization approach that incrementally constructs a summary by predicting key content selection and generating natural language phrases. Paulus et al. [27] presented a deep reinforced model for abstractive summarization that incorporates a reinforcement learning framework to train the model to generate high-quality summaries.

Pasunuru et al. [26] focused on generating video captions but leveraged entailment rewards to reinforce abstractive summarization, improving the quality and informativeness of the generated captions.

Zhou et al. [34] outlines a neural document summary system that efficiently combines extractive and abstractive approaches to provide clear and detailed summaries. The algorithm jointly learns to score and pick phrases. See et al. [28] in their paper, explored a network-based pointer generator for summarization, which combines extractive and abstractive methods to generate summaries by copying content from the original data.

Liu et al. [19] studied how using BERT, a pre-trained encoder, for text summarization, achieves improved performance by leveraging the rich contextual representations. Liu et al. [18] introduced PreSumm, a neural model for abstractive text summarization that incorporates transformer-based architectures and achieves state-of-the-art performance on various summarization datasets. Ma et al. [21] explored the integration of external knowledge sources into pre-trained transformers for abstractive summarization, aiming to enhance the summary generation process with additional information.

Li et al. [15] proposed DRGAT, a dual-reading graph attention network, which leverages graph attention mechanisms to capture global and local dependencies for abstractive summarization.

Table-based summarization has also been addressed in many research papers. Arvind et al. [3] proposes a structure-aware sequence-to-sequence learning model for generating natural language text from tables. Lebret et al. [13] presents a neural text generation model that converts structured data, including tables, into natural language text, with a specific focus on generating biographical summaries.

Li et al. [14] proposes structured attention networks for table-to-text generation, which effectively capture dependencies between table elements and generate coherent and informative summaries. Krishrnamurty et al. [12] presents a neural semantic parsing model for semi-structured tables, enabling the conversion of table content into a structured representation that can be used for generating natural language summaries.

Dong et al. [5] provides an overview of neural text generation techniques in structured data-to-text applications, including table-based summarization, discussing challenges and potential solutions. Agarwal et al. [9] proposes an effective hierarchical encoder that leverages structural information in tables for table-to-text generation, improving the quality and coherence of

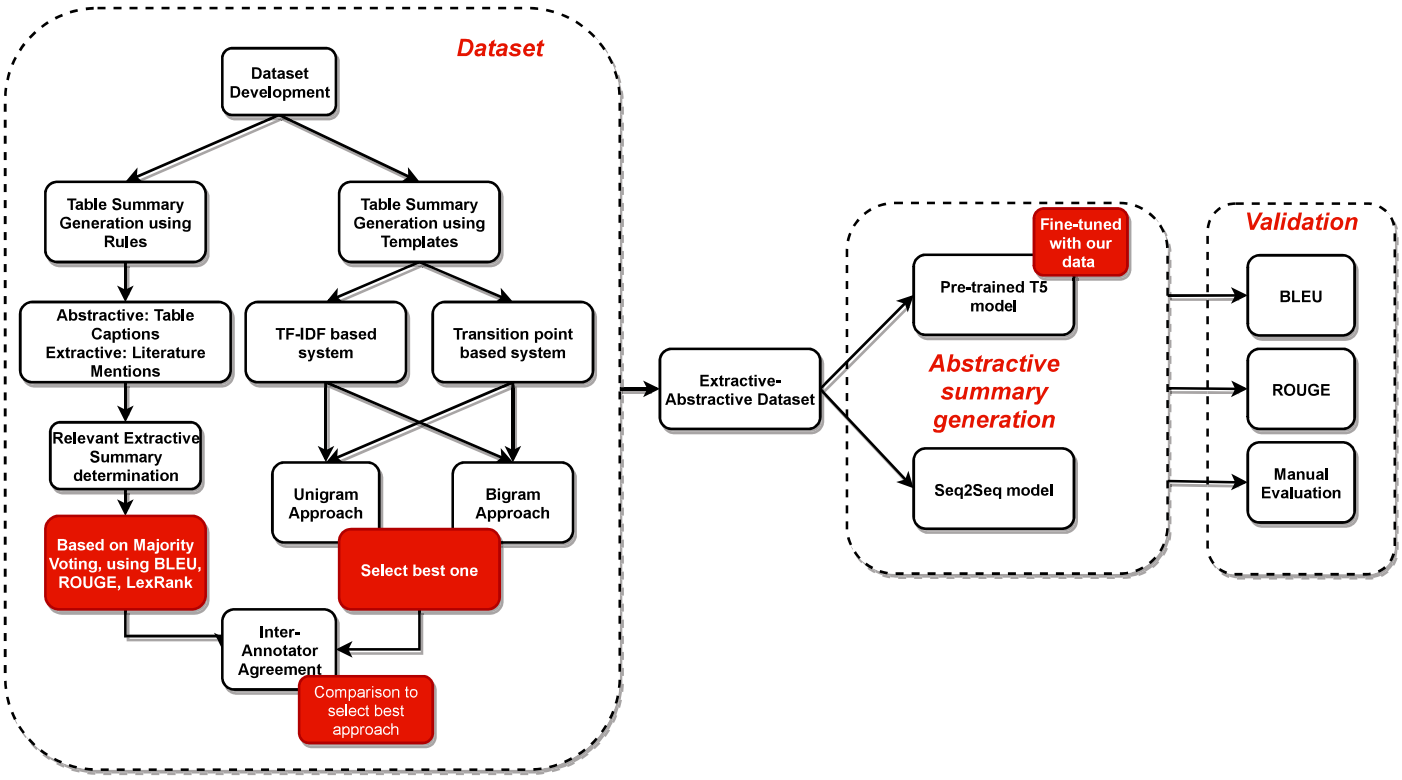

Figure 1: Flow of the data extraction, summary generation and testing process

generated summaries.

Gao et al. [7] introduces Table2Seq, a neural sequence generation model for table-to-text tasks, employing a novel column-aware attention mechanism to capture important table structure information. Lin et al. [17] presents hybrid pointer generator networks for table-to-text generation, incorporating both extractive and abstractive methods to generate coherent and accurate summaries. Cui et al. [4] proposes a descriptive sentence generation model for table structure, enabling the generation of natural language descriptions that capture the essence of tabular data.

Ma et al. [22] provides a structure-aware convolutional seq-to-seq network to efficiently capture the hierarchical structure of tables to produce consistent and useful summaries. Zhang et al. [32] introduces a table-based neural text generation model that incorporates semantic constraints to guide the generation process and improve the coherence and fidelity of generated summaries. Jiang et al. [11] proposes a structure-aware transformer model for table-to-text generation, effectively capturing table structure information to improve the quality and coherence of generated summaries.

Zheng et al. [33] proposes a multimodal framework for table-to-text generation, which leverages both extractive and abstractive methods to generate summaries, taking into account both table content and associated textual descriptions.

As discussed above a lot of work has been done in the abstractive summarization field. However, none of these works addresses the challenge of developing scientific table summarization systems. In our work, we have addressed the challenge of developing a corpus containing extractive and abstractive summaries of scientific tables. We have also proposed systems for dataset development as well as extractive to abstractive summary generation for tables.

3 Dataset Development

As mentioned the absence of a dataset made it necessary for us to first develop a dataset. To construct this corpus, we obtained scientific articles from digital libraries as they typically include tables that present important information about research findings. We gathered 1,500 papers across 20 distinct domains in computer science, such as Automatic Summary, Machine Learning, and Machine Translation. Each article has an average of approximately 250 sentences, not including titles, author names, and section headings. We then employed two approaches to prepare our dataset as described in the following sections.

Dataset Preprocessing: Preprocessing is an important step in any data analysis task, as it can improve the quality and usability of the data, and make it easier to extract insights and information from it. This involves cleaning the data, transforming it into a suitable format, and extracting the relevant information without any errors or inconsistencies. Next features were extracted from the baseline format and this information was organized in a way

Table 1: Dataset statistics

Paper Type	#Tables	Type: Text	Type: Numeric	ESummary		ASummary
				#Eavg	Elen_avg	Alen_avg
Automatic Summary	510	130	370	3	16	11
Machine Learning	700	373	327	4	18	12
Machine Translation	420	150	280	3	16	10
Named Entity Recognition	789	553	236	2	16	14
Question Answering	553	120	433	3	15	13
Sentiment Analysis	421	125	296	2	14	14
Speech Recognition	700	432	286	5	13	13
Text Classification	567	265	302	3	15	15
Text Segmentation	700	432	268	2	13	13
Word Sense Disambiguation	650	324	326	1	11	13
Total no. of papers	1,500					

that makes it easier to work with in the upcoming sections.

3.1 *Table Summary Generation Using Rules*

Caption Identification: To provide a clear representation of the data presented in a table, a well-written caption is crucial. Captions may vary depending on the domain and writing style, so we developed a method to distinguish caption sentences from other sentences in the document. We discovered that captions in various papers consist of four parts: ⟨TABLE⟩, which is the word "Table", followed by ⟨INTEGER⟩, an integer indicating the table number in the paper. The integer is then followed by a ⟨DELIMETER ⟩, which is a delimiter such as a period or a colon. Finally, we have ⟨TEXT⟩, which is a description of the content of the table. If a sentence follows this structure, we label it as a caption sentence and consider it as a summary of the table's contents. This approach enables us to handle the diversity of caption formats, resulting in coherent and informative summaries for each table.

Relevant Sentence Extraction: Although captions can effectively describe the contents of a table, studies have demonstrated that captions alone may not provide readers with a complete understanding of the information presented in a table. To address this limitation, we have observed that tables are referenced at least once in the corresponding scientific document. Therefore, we have developed a method to extract the reference text for a table to obtain a more comprehensive understanding of it.

The initial step was to segment the document text into sentences. To identify relevant sentences, we followed the same approach as for caption extraction, but we omitted the delimiter part. Additionally, we noted that sentences located near the reference sentence are useful in providing context for the table. We assigned scores to each sentence based on its proximity and distance to the reference sentence. If the distance was within a certain threshold length (+/-) 1, we considered the sentence important and included it in the summary.

By including relevant sentences in the summary, we can provide readers with a more complete understanding of the table and its context. This approach complements the captions and addresses the issue of insufficient information presented by captions alone.

As understood, a table can have multiple extractive summaries but only one abstractive summary. This process is described in Figure 2.

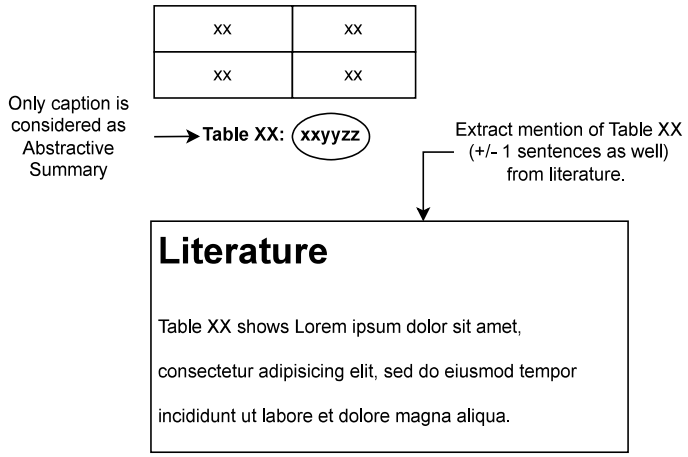

Figure 2: Process of caption identification and relevant sentence extraction

Annotation: After generating the abstractive and extractive summaries, we developed an annotated dataset to build a well-structured and easy-to-use corpus that can be automatically evaluated. We employed two separate approaches to evaluate the standard of our system-generated output as extractive and abstractive summaries which are discussed in the following subsections.

We also developed an output feature file using our system, which included additional features besides the summaries. These features included the paper ID, table ID, number of rows in a table, row attributes, column attributes, and the type of the table (numeric, text, or hybrid). By including these features, we aimed to provide more context about the tables and enable better analysis of the data.

Overall, the annotated dataset we constructed provides a comprehensive resource for researchers and practitioners in the field of computer science. The abstractive and extractive summaries and the additional features of the dataset make it an ideal resource for training and testing Natural Language Processing (NLP) models and for conducting further research in the field. Table 1 displays the characteristics of the corpus we developed.

Table 1 denotes the total number of tables in the dataset. Table

types can be text or numeric. ESummary denotes extractive summary and ASummary denotes abstractive summary. $\#Eavg$ is the average number of extractive summaries present per table per paper in a particular paper type.

Whereas, $Elen\_avg$ is the average length (in words) of an extractive summary in a particular paper type. $Alen\_avg$ on the other hand denotes the average length (in words) of an abstractive summary in a particular paper type.

3.1.1 Relevant Extractive Summary Selection. Extractive Summary Selection (ESS) is the process of selecting the most relevant extractive summary. It is crucial as it ensures that the model is trained on the best possible data and is more likely to produce accurate and informative summaries. Since our main aim is to ensure that the best quality extractive summary is selected for further works, we have used standard quality assessment tools like ROUGE (ESS_R), BLEU (ESS_B) and LEXRANK (ESS_L) and used a majority voting technique between them for selecting the most relevant extractive summaries.

It must be remembered that in the upcoming sections, the abstractive and the extractive summaries are considered as the reference and generated summaries respectively.

ESS_B: The BLEU score evaluates the accuracy of translations or summaries produced by computers in comparison to one or more references produced by humans. For every table i, the BLEU score between the abstractive summary and the extractive summaries for table i is calculated.

ESS_R: The effectiveness of automated summarization is evaluated using the ROUGE method. It determines how comparable the produced summary and the reference summary are based on the overlap of n-grams and their respective frequencies. The difference between the abstractive summary and the relevant extractive summaries for each table i is measured by the ROUGE score.

ESS_L: LexRank [6] is a graph-based algorithm for ranking sentences in a document based on their similarity to each other. It uses the concept of eigenvector centrality to score sentences based on their similarity to other sentences in the document. Sentences with high LexRank scores are considered to be the most important and relevant to the document. For every table i, the LEXRANK score of every $extractive_{ij}$ for table i is calculated.

Majority Voting Technique: Once the BLEU, ROUGE, and LEXRANK scores for each extractive summary $extractive_{ij}$ were obtained, we then wanted to select the most relevant and highly scored extractive summary. However, since the three metrics are different and have different ways of calculation, it was necessary to normalize the values first.

After normalizing, the majority voted summary by all the metrics was finally selected as the most relevant extractive summary as shown in the Equation (1), where Metric denotes either BLEU, ROUGE, or LEXRANK.

$$Relevant\_ES = MaxVoted(MAX(Metric(abstractive_i, extractive_{ij}) \tag{1}$$

3.1.2 Dataset Quality Evaluation. Initially in the dataset, an abstractive summary AB_1 had multiple extractive summaries E_1, E_2 mappings denoted by $AB_i \rightarrow E_j$, where i is the total number of abstractive summaries and j is the total number of extractive summaries for each i. However, after selecting the most significant extractive summary for each table as discussed in the previous sections, we have made the dataset more relevant and compact.

We have employed two methods for validating and evaluating the quality of the corpus namely, Inter Annotator agreement-based validation and Automatic Evaluation.

The following subsections provide a succinct overview of the evaluation methodology of the corpus.

Inter Annotator Agreement-based Validation: In order to validate this dataset, we employed two human annotators, A_1 and A_2, who were tasked with evaluating the mapping between an abstractive summary and the selected extractive summary for a particular table.

Each annotator was tasked to identify whether the mappings were valid according to their opinion. A valid mapping was given a score of "1" and an invalid mapping was given a score of "0". The dataset had 6,010 tables so the annotators were asked to validate a total of 6010 $AB_i \rightarrow E_j$ mappings.

Table 2 presents the confusion matrix constructed using the two annotators provided agreement-based scores for both of the labels (Valid - "1" and Invalid - "0").

With the help of these scores, we then calculate the agreement between annotators A_1 and A_2 using Cohen's Kappa[2] agreement analysis approach.

Cohen's Kappa coefficient score κ, which is defined in Equation (2) [31], which is used to illustrate the degree of agreement.

$$\kappa = \frac{Pr_a - Pr_e}{1 - Pr_e} \tag{2}$$

where Pr_a is the observed proportion of full agreement between two annotators. In addition, Pr_e is the proportion expected by a chance and so indicates a kind of random agreement between the annotators.

The final value of κ ranges from -1 to 1, with 1 denoting total agreement, -1 denoting complete disagreement, and 0 denoting agreement by chance.

The analysis of agreement using Cohen's Kappa, in this case, shows that for the abstractive to extractive mappings, the value

[2]https://en.wikipedia.org/wiki/Cohen's_kappa

Table 2: An inter annotator agreement analysis to validate the dataset

No. of Mappings ($AB_i \rightarrow E_j$) : 6,010		Annotator 1	
		Valid (Score =1)	Invalid (Score=0)
Annotator 2	Valid (Score =1)	5175	40
	Invalid (Score=0)	45	210
Kappa Score		0.824	

of κ is 0.824 with an agreement of 95% confidence interval. A higher κ value indicates a stronger agreement.

The purpose of this experiment was to assess the effectiveness of the proposed method in accurately identifying the summary of table content in a given document.

Automatic Evaluation: We employed two evaluation metrics, BLEU (Bilingual Evaluation Understudy) and ROUGE (Recall-Oriented Understudy for Gisting Evaluation), to further confirm the results of the external annotators. Based on n-gram matching, BLEU calculates the degree of similarity between a machine-generated summary and one or more reference summaries. The ROUGE family of assessment measures, on the other hand, focuses on the recall of significant data from the produced summary.

To do this, we selected all the 6010 $AB_i \rightarrow E_j$ mappings and calculated the BLEU and ROUGE scores of the extractive summary with respect to its abstractive summary. For this calculation, we used the AB_i as the reference summary and the most relevant extractive summary E_j as the candidate summary.

Table 3 reports the average BLEU and ROUGE-L (F1) scores for all combinations, while Figure 3 shows the BLEU scores that were obtained for the summary mappings for all possible combinations as mentioned above.

Similarly, Figure 4 depicts the Rouge scores that were obtained for the summary mappings for all possible combinations, viz. (i) *Both annotators agree*, (ii) *A1 agrees, A2 disagrees*, (iii) *A1 disagrees, A2 agrees* and (iv) *Both annotators disagree*. We have taken 40 summary mappings as its the least number of mappings in the confusion matrix above.

After analyzing the charts, we can come to the conclusion that the BLEU and ROUGE scores of the sample mappings that were agreed as VALID by both the annotators have higher values than the other combinations.

This essentially supports our theory that the summary samples serve as the best ones when both expert annotators are in agreement, demonstrating the datasets' quality.

3.2 Summary Generation Using Templates

Along with the dataset corpus using rules as discussed in the previous section, we have also proposed the development of two models that develop summary templates representing the extractive summary of a table. These templates can be utilized by any NLP researcher to study and develop more accurate and coherent summaries and also to generate abstractive summaries.

To do this, our system first identifies a set of significant terms

Table 3: An inter annotator agreement analysis to validate the dataset

No. of Mappings : 6,010			
Both Agree		A2 Agree	
Avg_BLEU	Avg_ROUGE-L	Avg_BLEU	Avg_ROUGE-L
94.1	0.79	49.5	0.40
A1 Agree		Both Disagree	
Avg_BLEU	Avg_ROUGE-L	Avg_BLEU	Avg_ROUGE-L
48.25	0.41	5.12	0.05

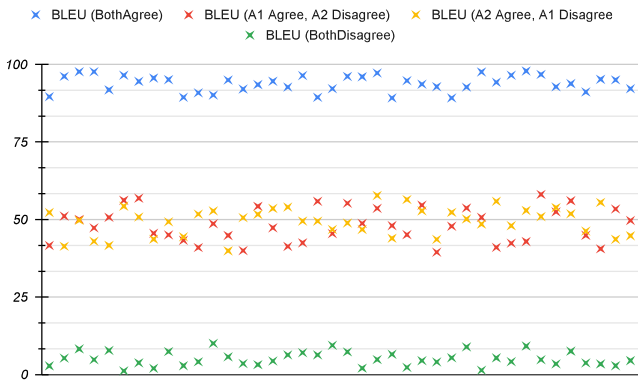

Figure 3: Inter-annotator BLEU scores for rule-based approach

from each scientific paper that is downloaded. These terms are then utilized to produce an extractive summary of the tables contained in the paper.

To assess the quality of the summaries in the dataset, we employ automatic as well as manual evaluation techniques.

The two systems that we have developed are explained in the subsection below.

3.2.1 TF-IDF. Under this system, we have proposed two approaches namely Unigram and Bigram approach to extract templates for extractive summary.

Unigram Approach: The corpus has a total of 1,500 papers, and for each table in each paper, we calculate the TF-IDF score of all the terms, excluding stop words, non-alphanumeric characters, and unnecessary punctuations.

We only consider terms that are both within the set of unique words and belong to the highest-scored terms for the template. This set of terms is referred to as the Template for Match (TS). While each table can have multiple extractive summaries, there is only one TS for all the summaries of a particular table. Therefore, we rank the summaries to determine which extractive summary matches best with the TS.

Bigram Approach: Another approach called the Bigram approach is developed, which considers a pair of consecutive words instead of single words. In this approach, the TF-IDF score is calculated for each bigram in the document. The

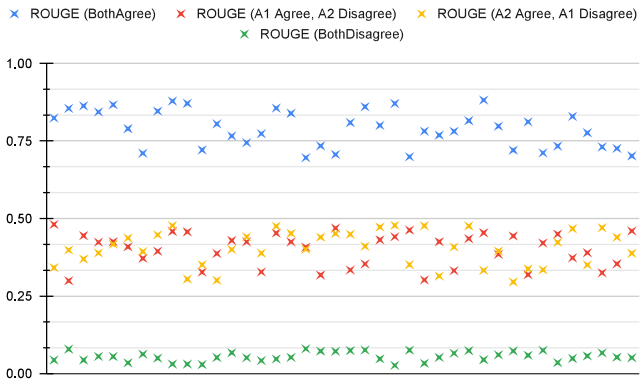

Figure 4: Inter-annotator ROUGE scores for rule-based approach

metrics used for selecting terms in the Template for match (TS) are BLEU and ROUGE. All of these scores are considered background knowledge in the textual entailment method, which is used for ranking the extractive summaries.

3.2.2 Transition Point Based System. Transition Point (TP) is a frequency number that separates the vocabulary of a text into two categories, low frequency, and high-frequency phrases. We have used TP to show how effective it is in indexing text because the mid-frequency phrases are so closely related to a document's conceptual substance. The Transition point system uses two methods: unigram and bigram.

Unigram Approach: A $document_i$ and its vocabulary $V_i = \{(w_j, tf_i(w_j)) | w_j \in D_i\}$, where $tf_i(w_j) = tf_{ij}$, and TP_i be the transition point of D_i. A collection of significant keywords that effectively represent the content of the document D_i using the following method.

$$R_i = w_j \mid ((w_j, \mathrm{tf}ij) \in V_i), (TPi \cdot (1-u) \leq \mathrm{tf}ij \leq TPi \cdot (1+u)) \tag{3}$$

The value of u lies between 0 and 1. Empirical evidence from experiments presented by Urbizagástegui [29] suggests that a threshold value of u=0.4 is effective. The transition point, TP, is calculated using the following formula:

$$TP = \frac{(-1+\sqrt{8 \times I_i})}{2} \tag{4}$$

The value of I_1 denotes the count of words that occur only once in the document. The terms whose frequencies lie in proximity to the transition point (TP) are considered significant and are given a higher weight for summarization purposes, while the remaining terms are assigned a weight close to zero.

Bigram Approach: The method described above for selecting important terms was enhanced by incorporating words with similar characteristics. This was achieved through the use of a co-occurrence bigram formula described in the work by López et al. [20]. The system based on this bigram approach was divided into three subsystems, namely Left Approach, Right Approach and Left-Right approach. Specifically, given a document D_i consisting only of terms selected using the TP unigram approach (c), the new important terms for D_i were obtained in different ways for the three subsystems. To achieve this, the bigram of each document was taken, and the TF score was calculated as the number of times the bigram occurred in the document. This approach enabled the identification of important bigrams that provided additional context and meaning to the document.

Left Approach: This approach examines the bigram's TF score, selecting only those with a value greater than one. For each of these bigrams, if the term from R_i appears in the rightmost position, the left term is selected as the new term to be added to D_i.

Right Approach: This approach focuses on the right-most term of the bigram when the terms in R_i appear in the left-most position.

Left-Right Approach: The final approach is a combination of the first and second approaches, where both left and right terms are considered if they appear in the bigram and satisfy the condition of having a minimum frequency of two.

3.2.3 Dataset Quality Evaluation. ***Automatic Evaluation:*** The experimentation results after applying automatic evaluation methods like BLEU and ROUGE to the output of the above-mentioned approaches are reported in Table 4.

After developing template-based summaries, it was noticed in the experimentation that by varying the number of terms for Template for Matching (TS), we get different BLEU and ROUGE scores for different numbers of terms. It was further noticed that BLEU scores increased with a smaller number of terms however, there was a decrease in the ROUGE score as the number of terms increase as seen in Table 4.

If the TF-IDF Unigram and Bigram approaches are compared, it is very clear that TF-IDF Unigram approach has better performance in extracting the summary templates.

Moreover, if the Transition Point Approach is considered, the scores are extremely lower than the other approaches.

The results of the automatic evaluation are depicted in Figures 5 , 6, 7 and 8.

If we notice Figures 5, 6, and 7, the observed disparity in the chart emphasizes the importance of agreement between annotators, as it directly impacts the quality and accuracy of the summaries. Thus, when both annotators reach a consensus on the validity of a summary, it can be seen that the summary in question exhibits a stronger resemblance to the summaries, as evidenced by its elevated ROUGE and BLEU scores.

Thus after using automatic evaluation measures, it can be

Table 4: Automatic evaluation scores for summary templates

TF-IDF Unigram Approach				
Terms	**BLEU**	**ROUGE-L**		
		Precison	**Recall**	**F-Measure**
10	46	0.26	0.71	0.34
20	40	0.53	0.60	0.51
30	36	0.72	0.50	0.53
TF-IDF - Bigram Approach				
Terms	**BLEU**	**ROUGE-L**		
		Precison	**Recall**	**F-Measure**
10	0	0.003	0.009	0.004
20	0	0.003	0.009	0.004
30	0	0.002	0.001	0.003
Transition Point System				
Bigram-Approaches	**BLEU**	**ROUGE-L**		
		Precison	**Recall**	**F-Measure**
	0.044	0.14	0.17	0.08
Left Approach	0.08	0.11	0.16	0.21
Right Approach	0.11	0.19	0.21	0.22
Left-Right Approach	0.13	0.14	0.02	0.12

concluded that the TF-IDF unigram approach performed better overall.

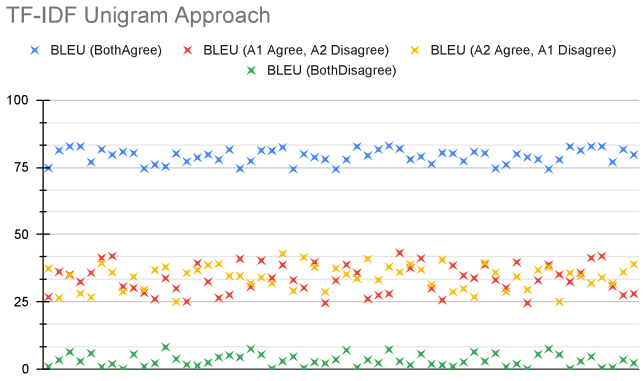

Figure 5: Inter-annotator BLEU scores for TF-IDF unigram Approach

Inter Annotator Agreement-based Validation: In order to validate further, we employed two human annotators, A_1 and A_2, who were asked to evaluate the quality of the summary templates obtained by the TF-IDF unigram approach. We selected only the TF-IDF unigram approach as it was clearly noted that it outperforms all the other approaches.

Each annotator was tasked to identify whether the mappings were valid according to their opinion. A valid mapping was given a score of "1" and an invalid mapping was given a score of "0". The dataset had 6,010 tables so the annotators were asked to validate a total of 6010 $AB_i \rightarrow E_j$ mappings.

Table 5 presents the confusion matrix constructed using the two annotators provided agreement-based scores for both of the labels (Valid - "1" and Invalid - "0").

With the help of these scores, we then calculate the agreement

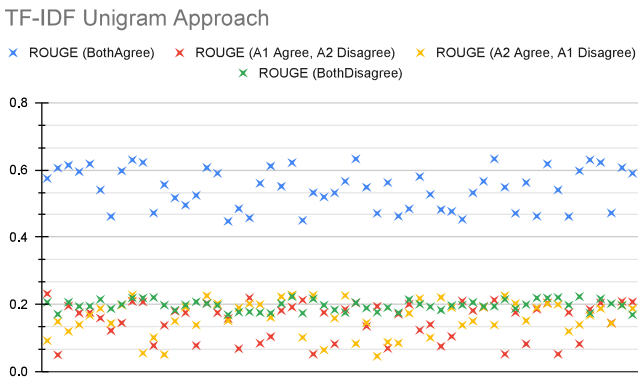

Figure 6: Inter-annotator ROUGE scores for TF-IDF unigram Approach

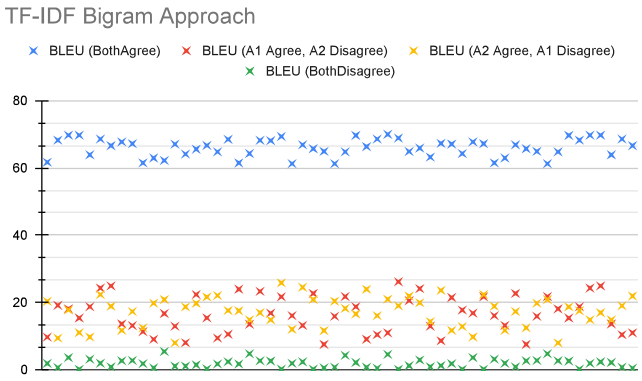

Figure 7: Inter-annotator BLEU scores for TF-IDF bigram Approach

between annotators A_1 and A_2 using Cohen's Kappa agreement analysis approach.

The analysis of agreement using Cohen's Kappa, in this case, shows that for the abstractive to extractive mappings, the value of κ is 0.568.

Comparison: Table 6 shows the comparison between the rule-based approach and template-based TF-IDF approach. In both cases, we have only taken those summary pairs which have been marked as valid summaries by both the external annotators. By observing the BLEU, ROUGE-L, and agreement scores we

Table 5: An inter annotator agreement analysis to validate the summary templates

No. of Mappings ($AB_i \rightarrow E_j$) **: 6,010**		Annotator 1	
		Valid (score = 1)	**Invalid (score = 0)**
Annotator 2	Valid (score = 1)	4,010	467
	Invalid (score = 0)	510	1,023
Kappa score	0.568		

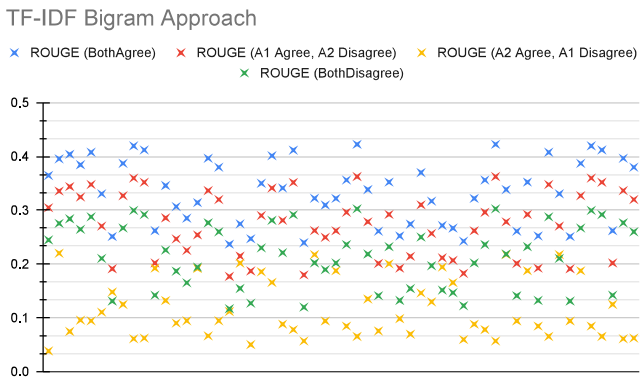

Figure 8: Inter-annotator ROUGE scores for TF-IDF bigram Approach

Table 6: Comparison between rule-based and TF-IDF-Unigram approach where both annotators have agreed

Rule Based Summary_Both Agree		
Avg_BLEU	**Avg_ROUGE-L**	**Kappa_Score**
94.1	0.79	0.824
Template Based_TF-IDF_Unigram_Both Agree		
Avg_BLEU	**Avg_ROUGE-L**	**Kappa_Score**
78.7	0.54	0.568

can easily conclude that the rule-based approach of dataset development is better than the template-based approaches. For this reason, we will be considering the summary generated by the rule-based approach in our next tasks.

4 Extractive to Abstractive Summary Generation

In this section, we propose models to develop abstractive summaries from relevant extractive summaries from the dataset.

Abstractive summarization has the potential to generate summaries that capture the underlying meaning of the text and can provide a more coherent and readable summary. One approach to generating abstractive summaries is to use extractive summaries as a starting point.

Using extractive summaries as a starting point for generating abstractive summaries has several benefits. Firstly, it can reduce the complexity of the summarization task. By using extractive summaries, the abstractive summarization system can focus on generating new sentences that capture the essence of the original document, rather than trying to capture all the details in the original document. Secondly, it can improve the quality of the generated summary. Extractive summaries provide a summary of the main ideas and key concepts in the original text, which can guide the generation of new sentences. This can result in a more coherent and informative summary than generating the summary from scratch. Finally, using extractive summaries can also improve the efficiency of the summarization process. Extractive summarization is a less computationally intensive task than abstractive summarization. By using extractive summaries as a starting point, the abstractive summarization system can reduce the amount of processing required to generate the final summary.

4.1 Abstractive Summary Generation

In general, the process of extractive text summarization involves two steps: sentence scoring and sentence selection. The first step involves determining the importance or relevance of each sentence, while the second step involves ranking the sentences based on their scores and selecting the most important ones to form the summary.

However, in this work, the process has been simplified even further. We have developed a dataset consisting of tables, along with their extractive and abstractive summaries. This means that instead of having to score and select individual sentences, we can focus on developing models to select the most relevant extractive summary for each table.

By using this approach, the researchers are able to leverage the information contained in the tables themselves, as well as the extractive summaries that have already been developed. This makes the process of summarization more efficient and accurate, as the models are able to focus on identifying the most important information in the tables and selecting the extractive summary that best captures this information.

Thus by incorporating additional sources of information and using pre-existing summaries as a starting point, we are able to develop more accurate and efficient summarization models.

In this section, we propose two separate tasks. The first task is the selection of the most relevant extractive summary and the second task is generating an abstractive summary by taking the selected extractive summary as the starting point.

In this task, our input is the carefully selected extractive summary from the previous task and the output is an abstractive summary. We have employed two approaches namely, the T5 model and Seq to Seq Model.

4.1.1 T5 Small Model. T5, which stands for "Text-to-Text Transfer Transformer", is a powerful NLP model developed by Google. It is based on the Transformer architecture, specifically designed for various text-based tasks, including text summarization.

The T5 model can take input in the form of text and generate output in the form of text as well. It follows a "text-to-text" approach, where the input and output are both represented as text strings.

In our work, we utilized T5 small model for generating an abstractive summary. The inputs given to the model are the selected extractive summary for each table.

The steps followed in utilizing this model are as follows:

Pre-processing the Text: In this step, we went through the input text and ensured that it is in a suitable format for the T5 model. This involves removing unnecessary details, formatting

the text, and performing any other necessary adjustments.

Fine-tuning the Model: Despite the fact that this is a pre-trained model, we have to fine tune it using our training dataset to make sure the output is relevant to the task at hand. For this purpose, we utilized 4,800 data samples ($AB_i \rightarrow E_i$), where AB_i is the abstractive summary and E_i is the selected relevant extractive summary using the majority voting technique as discussed in the previous sections. The training input was the extractive summary and the output was the related abstractive summary. We have used a dropout rate of 0.1 and a constant learning rate of 0.001. In this task, we experiment with input lengths of 4096 and 8192 and output lengths of 512.

Summary Generation: Once the model was trained we tested the model using the remaining 1,210 selected extractive summary samples (AB_i) and generated an output summary.

It must be noted that in order to instruct T5 to perform summarization, we had to prepend the input text with a specific task description. For example, we added "summarize:" to indicate that the model should generate a summary. The formatted input would then be "summarize: [input text]."

4.1.2 Seq to Seq Model. We decided to employ the sequence-to-sequence (seq2seq) model, which builds on the notion of sequence learning using neural networks, to produce abstractive summaries from extractive ones. Basically, the model accepts a sequence

$$X = \{x_1, x_2, ..., x_n\} \tag{5}$$

as input and attempts to produce the target sequence

$$Y = \{y_1, y_2, ..., y_m\} \tag{6}$$

as output.

where the input and target symbols, respectively, are x_i and y_i. The encoder and the decoder are the two components that make up the seq2seq model's architecture. We experimented with word-level embedding in a manner similar to the work of [23], and our model utilized the seq2seq architecture. Figure 9 illustrates how the seq2seq architecture operates at the word level. We used the Keras library for implementing the model.

Encoder: We utilized LSTM cells in the encoder design. One hot tensor of word-level embedded extractive summaries served as the cell's input. The outputs from the encoder were deleted but the internal states of each cell were kept. This is done in order to maintain context-level information. The decoder cell was then given these states as beginning states.

Decoder: An LSTM cell was once more used for building the decoder, with initial states as the encoder's hidden states. Sequences and states can both be returned by it. Williams' (1989) theory of "teacher forcing" learning was applied in this

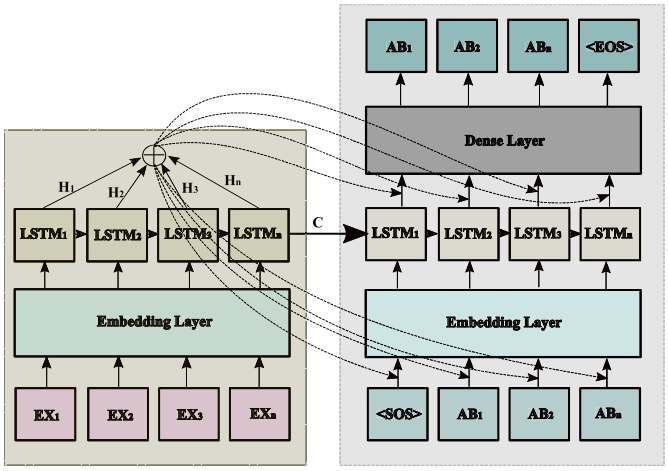

Figure 9: Architecture of the seq2seq model used to generate abstractive summaries from extractive summaries

instance. One hot tensor of abstractive summaries (embedded at the word level) served as the decoder's input, while the target data was the same but with an offset of one-time step ahead of the input. From the initial states that the encoder passed on, the information for generation is acquired. As a result, the decoder has the ability to produce target data [t+1,...] conditioned on the input sequence given targets [..., t]. One word is predicted for each output time step, thus predicting the output sequence.

Attention: The majority of computational neuroscience's work has focused on brain functions like attention [30]. This concept is based only very loosely on how people concentrate their visual attention. It is no longer necessary to encode the complete source sentence into a fixed-length vector thanks to the attention mechanism. Instead, we allowed the decoder to concentrate on various aspects of the source text at each stage of output production. In essence, we let the model decide what to *attend* based on the input sequence and the prediction's current status.

The context vector c_t is calculated mathematically at each time step t as a weighted sum of the source hidden states,

$$c_t = \sum_{t=1}^{Tx} \alpha_t h_t \tag{7}$$

Each attention weight α_t represents how much relevant the t^{th} source token x_t is to the t^{th} target token y_t and is computed as :

$$\alpha_t = \frac{1}{Z} exp(score(E_y(y_t - 1), s_{t-1}, h_t)) \tag{8}$$

where

$$Z = \sum_{k=1}^{Tx} exp(score(E_y(y_t - 1), s_{t-1}, h_k)) \tag{9}$$

The normalization constant is called Z. The function score() calculates the degree to which the source symbol T_x and the target symbol y_t match using a feed-forward neural network

with a single hidden layer. The target hidden state is represented by s_t and the target embedding lookup table by E_y.

For training the model, *batch size* was set to 64, *number of epochs* was set to 100, *activation function* was softmax, *optimizer* chosen was rmsprop and *loss function* used was sparse categorical cross-entropy. The learning rate was set to 0.001.

4.2 *Evaluation*

We tested the performance of the T5 model using ROUGE and BLEU metrics as they are the most standard summarization metrics available. Additionally, we have also evaluated the quality of the dataset by two other metrics namely adequacy and fluency [10], with the help of two linguists, familiar with the English language. The average values of the metrics are also reported. Adequacy shows how much of the meaning of the source summary is expressed in the generated abstractive summary. Whereas, fluency depicts how well-formed the generated summary sentence is grammatically and can be easily understood by a native speaker. Fluency and adequacy are measured in the range of 1 to 5, where 1 represents the lowest value while 5 represents the highest.

Table 7 shows the results. As observed from the results, we can see that the pre-trained fine-tuned T5 model has performed better than the Seq-to-Seq model. This is because the size of the training and testing dataset was less and hence the model could not be trained adequately.

Table 7: Comparison between T5 and seq-to-seq model

Metrics	Fine-tuned T5 Model	Seq-to-Seq Model
Avg_BLEU	58.2	16.25
Avg_ROUGE-1	0.36	0.21
Avg_ROUGE-L	0.31	0.18
Avg_Adequacy	4.02	2.07
Avg_Fluency	3.91	1.83

5 Discussion and Future Scope

From the results in Table 4, we have clearly seen that the TF-IDF unigram approach outperforms the other two approaches by a substantial margin which paved the way for us to proceed the next tasks with the TF-IDF unigram approach outputs. As a future scope, we can experiment with the outputs of the other approaches as well and make a comparison after all tasks are done.

In addition, from the results shown in Table 7, we can easily see that the Fine tuned T5 model outperforms the Seq-to-Seq model by quite a margin. Though the margin looks small for some parameters like Rouge-1 it is quite substantial concerning the abstractive summary output. This is mainly due to the fact that the dataset that we developed is not sufficient enough for a deep learning model. Thus, as a future prospect, we can aim to increase the size of the summary corpus so that the models can be trained more efficiently to generate coherent summaries.

Furthermore we can aim to introduce a new graph based neural network model, as graph summaries may provide a more succinct representation of an input sentence. Since, a graph-based network's main goal is to maintain predefined properties that are important for particular tasks, including queries on the output summary, it can be highly beneficial for our task as well.

6 Conclusion

In this work, we have addressed the challenges that occur while trying to summarize tables in scientific papers. The data required for training such scientific table summarization systems are very scarce and hence we have proposed the development of a high-quality corpus consisting of both extractive and abstractive summaries. We have proposed two approaches for the same; rule-based and template based. In the rule-based approach we have taken the hypothesis that the caption of the table is its abstractive summary and the citation sentences of the table in the paper are its extractive summaries. The dataset evaluation results clearly depict that the rule-based approach generates much better summaries than the template-based approach. Subsequently, two models, fine-tuned T5-small and seq-to-seq models were also proposed to generate abstractive summaries from extractive ones. The results clearly show that the pre-trained T5 model performs better than the seq-to-seq model.

References

[1] M. Allahyari, S. Pouriyeh, M. Assefi, S. Safaei, E. D. Trippe, J. B. Gutierrez, and K. Kochut. Text Summarization Techniques: A Brief Survey. *arXiv preprint arXiv:1707.02268*, 2017.

[2] C. An, M. Zhong, Y. Chen, D. Wang, X. Qiu, and X. Huang. Enhancing Scientific Papers Summarization with Citation Graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 35:12498–12506, 2021.

[3] V. Arvind and B. Ravindran. Table-to-text Generation by Structure-aware Seq2Seq Learning. 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 4881–4888, 2019.

[4] Y. Cui, Z. Chen, W. Che, and T. Liu. Table2Text: Descriptive Sentence Generation for Table Structure. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1421–1431, 2020.

[5] L. Dong, F. Wei, and C. Tan. Neural Text Generation in Structured Data-to-Text Applications. In *(EMNLP-IJCNLP)*, pages 1321–1331, 2019.

[6] G. Erkan and D. R. Radev. Lexrank: Graph-Based Lexical Centrality as Salience in Text Summarization. *Journal of Artificial Intelligence Research*, 22:457–479, 2004.

[7] T. Gao, C. Zhu, K. Zhang, and X. Ren. Table2Seq: Neural Sequence Generation for Table-to-Text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1323–1324, 2019.

[8] S. Gehrmann, K. Deng, and A. M. Rush. Bottom-Up Abstractive Summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4098–4109, 2018.

[9] F. X. Q. B. Gong, Heng and T. Liu. Table-to-Text Generation with Effective Hierarchical Encoder on Three Dimentions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 3143–3152, 2019.

[10] M. Hearne and A. Way. Statistical Machine Translation: A Guide For Linguists and Translators. *Language and Linguistics Compass*, 5(5):205–226, 2011.

[11] L. Jiang, M. Yu, M. Zhou, and G. Neubig. Better Structure-aware Transformer for Table-to-Text Generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 45–67, 2021.

[12] J. Krishnamurthy, P. Dasigi, and M. Gardner. Neural Semantic Parsing with Type Constraints for Semi-Structured Tables. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1516–1526, 2017.

[13] R. Lebret, D. Grangier, and M. Auli. Neural Text Generation from Structured Data with Application to the Biography Domain. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1203–1213, 2016.

[14] C. Li and W. Lam. Structured Attention Networks for Table-to-Text Generation. *ACM Transactions on Information Systems (TOIS)*, 38(2):1–27, 2020.

[15] Z. Li, X. Liu, Y. Chen, and M. Sun. DRGAT: A Dual-Reading Graph Attention Network for Abstractive Summarization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 142–156, 2021.

[16] C.-Y. Lin. Rouge: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, pages 74–81, 2004.

[17] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu. Hybrid Pointer Generator Networks for Table-to-Text Generation. *ACM Transactions on Information Systems (TOIS)*, 38(3):1–27, 2020.

[18] Y. Liu and M. Lapata. PreSumm: A Neural Model for Abstractive Text Summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3142–3152, 2020.

[19] Y. Liu, M. Lapata, and F. Li. Text Summarization with Pretrained Encoders. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 3730–3740, 2019.

[20] F. R. López, H. Jiménez-Salazar, and D. Pinto. A Competitive Term Selection Method for Information Retrieval. In *International Conference on Intelligent Text Processing and Computational Linguistics, Springer*, pages 468–475, 2007.

[21] Q. Ma and S. Sakti. Incorporating External Knowledge into Pre-trained Transformers for Abstractive Summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1–13, 2020.

[22] Z. Ma, J. Li, G. Zhou, and J. Su. Structure-Aware Convolutional Sequence-to-Sequence Model for Table-to-Text Generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 112–125, 2020.

[23] S. K. Mahata, A. Garain, A. Rayala, D. Das, and S. Bandyopadhyay. A Hybrid Approach To Machine Translation For Lithuanian To English. *2019 Fourth Conference on Machine Translation*, pages 283–286, 2019.

[24] R. Nallapati, F. Zhai, and B. Zhou. Abstractive Text Summarization using Sequence-to-Sequence RNNs and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL)*, pages 280–290, 2016.

[25] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[26] R. Pasunuru and M. Bansal. Reinforced Video Captioning with Entailment Rewards. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 979–985, 2017.

[27] R. Paulus, C. Xiong, and R. Socher. A Deep Reinforced Model for Abstractive Summarization. In

Proceedings of the 6th International Conference on Learning Representations (ICLR), pages 1–13, 2018.

[28] A. See, P. J. Liu, and C. D. Manning. Get To The Point: Summarization with Pointer-Generator Networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1073–1083, 2017.

[29] R. Urbizagástegui. Las Posibilidades de la Ley de Zipf en la IndizaciON AutomATica. *Reporte de la Universidad de California Riverside*, 1999.

[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is All You Need. In *Advances in Neural Information Processing Systems*, pages 6000–6010, 2017.

[31] A. J. Viera and J. M. Garrett. Understanding Interobserver Agreement: The Kappa Statistic. *Fam med*, 37(5):360–363, 2005.

[32] Y. Zhang, Y. Zhang, Z. Wang, and L. Huang. Table-based Neural Text Generation with Semantic Constraints. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 102–113, 2021.

[33] Y. Zheng, X. Ye, Z. Lin, and Z. He. Extractive or Abstractive? A Multimodal Framework for Table-to-Text Generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1–20, 2021.

[34] Q. Zhou, N. Yang, F. Wei, C. Tan, H. Bao, and M. Zhou. Neural Document Summarization by Jointly Learning to Score and Select Sentences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 654–663, 2018.

Monalisa Dey is an Assistant Professor at the Institute of Engineering and Management, Kolkata (IEM) in the Computer Science and Engineering Department. She is currently pursuing her Ph.D. from Jadavpur University. In her previous roles, she was associated with JIS College of Engineering, as an Assistant Professor. She has completed her B.Tech and M.Tech, in Computer Science and Engineering, from West Bengal University of Technology and NIT, Durgapur. She has more than 20 research papers in reputed journals and conferences.

Sainik Kumar Mahata is an Assistant Professor at the Institute of Engineering and Management, Kolkata (IEM) in the Computer Science and Engineering Department. He also shoulders administrative responsibilities in the capacity of Assistant Head of Department and Placement Coordinator. He has completed his Ph.D. from Jadavpur University. He was awarded the prestigious Visvesvarya Ph.D Fellowship during his Ph.D studies. He has strong proficiency in Python, Natural Language Processing and Machine Translation. In his previous roles, he was associated with JIS College of Engineering, and Narula Institute of Technology in the role of an Assistant Professor. He has more than 30 research papers in reputed journals and conferences.

Dipankar Das is an Assistant Professor in Computer Science and Engineering Department at Jadavpur University, Kolkata, India. He was a Young Faculty Research Fellow, Visvesvaraya PhD Scheme for Electronics IT, Media Lab Asia, Ministry of Electronics and Information Technology, India. He completed his Ph.D from Jadavpur Unievrsity. His areas of interest are Natural Language Processing, Social Networks, Sentiment/Emotion Analysis, Information Extraction, Ontology Engineering, Psycho linguistics, Machine Learning, Code-Mixing, Dialougue Management, Fake News etc. He has more than 90 research papers published in reputed conferences and journals.