ECTI Transactions on Computer and Information Technology
Journal homepage: https://ph01.tci-thaijo.org/index.php/ecticit/
Published by the ECTI Association, Thailand, ISSN: 2286-9131

SETA- Extractive to Abstractive Summarization with a Similarity-Based Attentional Encoder-Decoder Model

Monalisa Dey[1], Sainik Kumar Mahata[2], Anupam Mondal[3] and Dipankar Das[4]

ABSTRACT

Summarizing information provided within tables of scientific documents has always been a problem. A system that can summarize this vital information, which a table encapsulates, can provide readers with a quick and straightforward solution to comprehend the contents of the document. To train such systems, we need data, and finding a quality one is tricky. To mitigate this challenge, we developed a high-quality corpus that contains both extractive and abstractive summaries derived from tables, using a rule-based approach. This dataset was validated using a combination of automated and manual metrics. Subsequently, we developed a novel Encoder-Decoder framework, along with attention, to generate abstractive summaries from extractive ones. This model works on a mix of extractive summaries and inter-sentential similarity embeddings and learns to map them to corresponding abstractive summaries. On experimentation, we discovered that our model addresses the saliency factor of summarization, an aspect overlooked by previous works. Further experiments show that our model develops coherent abstractive summaries, validated by high BLEU and ROUGE scores.

Article information:

Keywords: Summarization, TableSum, Extractive, Abstractive, Saliency

Article history:
Received: January 19, 2024
Revised: April 4, 2024
Accepted: May 23, 2024
Published: June 21, 2024
(Online)

DOI: 10.37936/ecti-cit.2024183.255552

1. INTRODUCTION

Summarization, the technique of compressing information into summaries while maintaining key elements, is an effective tool for controlling information overload [1]. Summarization uses two primary methods: extractive and abstractive.

Extractive summarization aims to preserve the original meaning by selecting significant phrases and sentences from the source document [2]. While this strategy is excellent at maintaining coherence and consistency, it may suffer from redundancy and coherence. In contrast, abstractive summarization creates new phrases and sentences to convey the essential themes in a more reduced fashion [3]. However, this strategy may have issues picking relevant vocabulary and assuring correctness.

Both extractive and abstractive summarization techniques have pros and cons, making them appropriate for various information types and situations. A practical summary requires striking a compromise between maintaining vital elements and delivering them succinctly.

This paper focuses on scientific publications, with a particular emphasis on the use of tables in presenting complex information. Tables provide particular problems to typical retrieval systems because they combine content and presentation elements [4]. As a result, it is critical to create a summary method that is specifically tailored to the content that tables include. Such a system would give researchers summaries, making it easier to understand crucial information without having to sift through the entire document.

However, there is a significant challenge for developing these systems: a shortage of suitable training datasets and evaluation algorithms [5, 6]. Training datasets are critical for training machine learning algorithms on how to adequately summarise using examples and patterns. Similarly, rigorous assessment methods are required to determine the correctness and efficacy of these summarization systems. The current lack of these resources hinders the development of systems that summarize tables. Overcoming this barrier necessitates coordinated efforts to de-

[1,2,3] The authors are with the Institute of Engineering and Management, Kolkata, India, E-mail: monalisa.dey.21@gmail.com, sainik.mahata@gmail.com and link.anupam@gmail.com
[4] The author is with the Jadavpur University, Kolkata, India, E-mail: dipankar.dipnil2005@gmail.com

velop comprehensive and coherent training datasets and also devise effective evaluation methodologies. By addressing these issues, researchers can create dependable table-summarizing systems that streamline information access and improves research workflows. Thus, keeping these issues in mind, the contributions of this paper are listed below.

The first contribution focuses on the problem of the scarcity of appropriate training and testing data for scientific table summarising systems. For this, we developed an extensive gold standard corpus TABLESum that contained carefully selected pertinent sentences from the paper's text that provided information about the table, as extractive summaries and captions of tables as abstractive summaries. The selection of relevant sentences from text bodies required the development of advanced techniques. These techniques ensure the careful extraction of pertinent data, while handling the difficulties in identifying and separating information inside scientific publications.

As discussed above, sentences referring to a table in the paper text and the table captions are the extractive and abstractive summaries for that table, respectively. Hence, the hypothesis is that for every table, there is one abstractive summary, but multiple extractive summaries. This leads to our subsequent contribution that involves the development of methods for selecting the most ideal extractive-abstractive summary pair for each table in the scientific publication. In order to achieve this, we employed a rule-based approach. To develop these rules, we used linguistic principles and domain-specific expertise to ensure that the rules adequately capture the substance of the original text. The validation step of our method consists of rigorous assessments using automated measures ROUGE [7] and BLEU [8], which are well-known for their effectiveness in measuring summarization quality. Furthermore, human evaluations provide significant qualitative insights into the subjective components of summary production.

Finally, to create an abstractive summary from the selected extractive summary, as our last contribution, we proposed the development of a novel seq-to-seq similarity-based attentional encoder-decoder architecture. This model includes extractive summaries with associated inter-sentential similarity embeddings, which are represented as an adjacency matrix of similarity scores computed using methods like as cosine similarity and Jaccard similarity. The proposed architecture takes as input the extracted summary for each table and generates an abstract representation of it. Our method prioritizes saliency, adequacy, fluency, and coherence. The table-based summarization systems developed earlier overlooked these factors. Our goal is to enhance summarization procedures by addressing these essential aspects. Figure 3 describes the architecture in detail. Our goal is to enhance summarization procedures by addressing these key aspects. Figure 3 provides a detailed description of the architecture flow.

The remainder of the paper is structured as follows: Section 2 discusses the recent developments in this area. Section 3 describes the dataset's development process. Section 4 describes the methods and models used to create extractive to abstractive summaries. Section 5 discusses the results. Finally, Section 6 presents the conclusion.

2. RELATED WORKS

The endeavor to summarize vast amounts of information has led to the development of two primary techniques: extractive and abstractive summarization [29]. Extractive summarization involves meticulously selecting pertinent sentences from the source document and presented collectively as the output. Conversely, abstractive methods take a different approach, generating new phrases. This section encompasses a review of literature covering the transition from extractive to abstractive summarization, along with table-based summarization techniques that underscore notable advancements in the field of study.

The most initial attempts at automatic summarization used extractive techniques, which locate words or phrases in the source document that contain the most relevant information. Some experiments involving encoder-decoder designs advanced such ideas even further. Nallapati *et al.* [9] and Jianpeng *et al.* [10] used encoder-decoder neural networks as binary classifiers to determine whether or not every sentence in a document should be included in the extracted summary. Chen *et al.* [11] use a pointer network to select sentences from the content that form the extracted summary.

The attention-based encoder-decoder paradigm has been extensively researched in abstractive summarization, motivated by the effectiveness of neural networks [12, 13]. Moreover, the benefits of extractive, abstractive, and attention-based models were initially combined together in [14, 15] which provided very good results. In table summarization however the challenge was always the scarcity of enough datasets. There are various table summarising or table-to-text generating datasets in the literature right now, such as WEATHERGOV [16], WikiBio [17], and so on, but none of them deal with scientific tables. Some sophisticated approaches, such as hierarchical-encoder [18] and lattice [19], performed well on these existing datasets. However, available datasets and approaches are often restricted to producing brief descriptions for a small number of cells. In fact, table-based summary has been discussed in several research articles. Arvind *et al.* [20] provide a structure-aware sequence-to-sequence learning technique for producing natural language text from tables. Li *et al.* [21] 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.* [22] 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.* [23] provides an overview of neural text generation techniques in structured data-to-text applications, including table-based summarization. Zheng *et al.* [24] 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 table summarization field. However, none of these works addresses the challenge of developing scientific table summarization systems. In our work, we have addressed this 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 PREPARATION- TABLESUM

As discussed earlier, we wanted to develop a high-quality corpus, containing both abstractive and extractive summaries that will help us in training a system that can generate abstractive summaries from extractive ones. Our collection of tables includes the summaries that go along with them, which we took out of scientific articles. This is a resource that hasn't been included in previous literature. A significant problem in the area is addressed by this new dataset, which is the dearth of suitable training and testing data for scientific table summarising systems. For this, we source scientific articles from digital libraries, and extracted the tables from them.

Over 2600 papers we collected which spanned 20 different domains in computer science like Machine Translation, Sentiment Analysis, Summarisation etc., to name a few. The quantitative details of the raw dataset are provided in Table 1. Entries like $ESummary$, denotes extractive summary, and $ASummary$ denotes abstractive summary.
$sharpEavg$, $Elen\_avg$, and $Alen\_avg$ represent the average number of extractive summaries per table, average amount of words in each extractive summary, and average number of words in each abstractive summary, respectively. These derived articles have an average of about 268 sentences.

The initial dataset pre-processing phase encompasses cleaning the data, converting it into an appropriate format, and extracting relevant information without errors or inconsistencies. Following this, features were derived from the baseline format, and the information was structured for ease of use in the subsequent sections. This led to the design of $TABLESum$ which is our gold standard dataset that can be used for generating table summaries. This entire process is discussed in the upcoming subsection 3.1. Next, Subsection 3.2 describes the process for identifying the most relevant extractive-abstractive summary pair and validating the dataset using automated and human evaluation methods.

3.1 Table Summary Generation

Caption Identification: Providing a clear depiction of data in a table requires developing a way to identify caption sentences from other content in a document. Effective information conveying relies heavily on captions, which can take many forms depending on the subject matter. After looking through a number of articles, we found that captions have a standard format that consists of four main components: the word "$< Table >$", an associated integer that indicates the table number, a delimiter, and a description that describes the contents of the table. We designate a sentence as a caption sentence when it follows this structured arrangement because we view it as a brief, gold standard summary of the table's contents. The hypothesis is that since the author has written it, hence it is accurate. The caption of the table is represented as its abstractive summary.

Table Relevant Sentence Extraction from Text: Although table captions do a good job of describing the contents of the table, they may be unable to give a full understanding. To navigate around this, we extract the corresponding table's reference text (text where the table is cited in the paper) and focus on the sentences that are close to the table reference. Relevant sentences are then included as the table's extractive summary by allocating scores to these sentences according to proximity.

In our work, we considered a sentence as significant and added it to the summary if the distance was within a predetermined threshold length $(+/-1)$. Thus by giving context, this approach improves comprehension of the summary. As indicated, there can be more than one extractive summary for a table, but only one abstractive summary.

After producing abstractive as well as extractive summaries, we carefully selected and annotated the dataset to create a coherent and intuitive corpus for automated evaluation. We employ two different methods in our evaluation process to evaluate the quality of output produced by the system, with an emphasis on extractive and abstractive summaries. The following subsection discuss in further depth about these assessments.

Table 1: *TABLESum Dataset Statistics.*

Paper Type	♯ Tab les	Ty pe: Te xt	Type : Num eric	ESummary		ASum mary
				♯Ea vg	Elen_ avg	Alen_ avg
Automatic Summary	895	347	548	3	16	11
Machine Learning	845	423	422	4	15	12
Machine Translation	689	268	421	3	16	10
Named Entity Recognition	956	632	324	2	16	14
Question Answering	925	434	491	3	15	13
Sentiment Analysis	650	275	375	2	14	14
Speech Recognition	598	277	321	5	13	13
Text Classification	955	431	524	3	15	15
Text Segmentation	652	414	238	2	16	13
Word Sense Disambiguation	650	324	326	1	14	13
Total No. of papers	2600					

3.2 Selection of Relevant Extractive Summary

Relevant Extractive Summary Selection (RES) entails selecting an extractive summary that is most closely related to the context or issue. 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* (RES_R), *BLEU* (RES_B) and *LEXRANK* (RES_L) [25]. Consequently we have used a majority voting technique between them for selecting the most relevant extractive summary. It must be remembered that in the upcoming sections, the abstractive and extractive summaries are considered the reference and generated summaries, respectively.

RES*$_B$*: 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.

RES*$_R$*: 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.

RES*$_L$*: LexRank 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 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.

$$Relevant_{ES} = MaxVoted(MAX(Metric\ abstractive_i, extractive_{ij})) \quad (1)$$

3.2.1 Validation of Dataset Quality

Initially in the dataset, an abstractive summary AB1 had multiple extractive summaries E1, E2 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.

Next 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

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 7815 tables so the annotators were asked to validate a total of 7815 $AB_i->E_j$ mappings. Table 2 presents the confusion matrix constructed using the two annotators provided agreement-based scores for both 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 agreement analysis approach. The Cohen’s Kappa coefficient score k, is defined in Equation 2 [26]. This score illustrates the degree of agreement.

$$k = \frac{Pr_a - Pr_r}{1 - pr_e} \qquad (2)$$

Where Pr_a is the observed proportion of complete agreement between two annotators. Furthermore, Pr_e is the proportion expected by chance, indicating a form of random agreement among the annotators. The final value of k 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 of k is 0.846 with an agreement of 96% confidence. A higher k value indicates a more substantial agreement.

Table 2: *An Inter-Annotator Agreement Analysis to Validate the Dataset.*

No. of Mappings (AB_i-$>E_j$): 7815		**Annotator 1**	
		Valid (score=1)	**Invalid (score=0)**
Annotator 2	Valid (score=1)	6953	110
	Invalid (score=0)	100	652
Kappa Score	**0.846**		

Automatic Evaluation

Next, 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 focuses on the recall of significant data from the produced summary.

To do this, we selected all the 7815 $AB_i->E_j$ mappings and calculated the BLEU and ROUGE scores of the extractive summary with 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 1 shows the BLEU scores obtained for the summary mappings for all combinations, as mentioned above. Similarly, Figure 2 depicts the Rouge scores obtained for the summary mappings for all 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 it is the least number of mappings in the confusion matrix above. After analyzing the charts, we can conclude that the BLEU and ROUGE scores of the sample mappings agreed as VALID by both the annotators have higher values than the other combinations.

This supports our theory that the summary samples serve as the best ones when both expert annotators agree, demonstrating the quality of the dataset.

Table 3: *An Inter-Annotator Agreement Analysis to Validate the Dataset.*

No. of Mappings: 7815			
Both Agree		**A2 Agree**	
Avg_BLEU	**Avg_ROUGE-L**	**Avg_BLEU**	**Avg_ROUGE-L**
93.1	0.65	51.5	0.45
A1 Agree		**Both Disagree**	
Avg_BLEU	**Avg_ROUGE-L**	**Avg_BLEU**	**Avg_ROUGE-L**
47.25	0.45	52.2	0.11

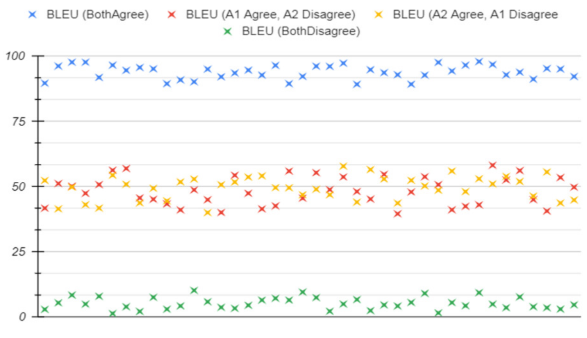

Fig.1: ***Inter-Annotator BLEU Scores.*** *This Figure displays BLEU scores for 40 summary mappings. Blue stars represent agreed-upon summaries, red/yellow stars denote disagreements by annotators A1/A2 and green stars signify mutual disagreement. Valid mappings show higher BLEU scores. Partial agreement yields average scores, while lack of consensus results in lower scores, indicating strong inter-annotator agreement.*

4. SETA - EXTRACTIVE TO ABSTRACTIVE SUMMARY GENERATION

In this section, we provide a seq-to-seq attention-based architecture that, after training on the produced dataset, would learn to generate abstractive summaries from relevant extractive summaries.

This model consists of two parts: an encoder and a decoder. The encoder component of the model processes a mix of extractive summaries and inter-sentence similarity embeddings. These embeddings are represented by an adjacency matrix filled with similarity scores. These scores are generated using various methods, including cosine similarity and Jaccard similarity.

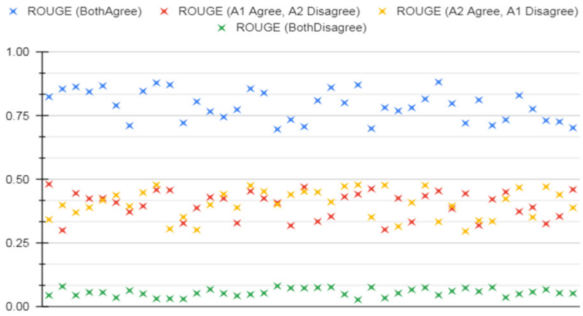

Fig.2: ***Inter-Annotator ROUGE scores.*** *This Figure displays ROUGE scores for 40 summary mappings. Blue stars represent agreed-upon summaries, red/yellow stars denote disagreements by annotators A1/A2 and green stars signify mutual disagreement. Valid mappings show higher ROUGE scores. Partial agreement yields average scores, while lack of consensus results in lower scores, indicating strong inter-annotator agreement.*

On processing, the encoder generates a context vector that contains knowledge of the whole sentence. This context vector initializes the decoder part, which acts as a language model that maps between abstractive summaries, differentiated by time frames , which in our case are t and $t + 1$. The description of the process is depicted below and shown in Figure 3.

4.1 Vectorisation of the summaries

To generate the vectors to feed as input to our developed model, the relevant extractive summary selected by majority voting technique is considered. As explained, this summary contains the reference sentence x, the sentence that comes before it $(x-1)$, and the sentence that comes after it $(x + 1)$. We did this to explore the hypothesis that the selected extractive summary (x) bears similarity to its preceding $(x - 1)$ and subsequent sentences $(x + 1)$.

Integrating these sentences as input is essential for generating context in the summary and ensuring it meets the criteria of saliency, non-redundancy, and fluency within a coherent and organized framework. We transform the extractive summary for every abstractive sample into an x X y vector, where y denotes the average word count value of the extractive summary samples. We decided the value of y using a histogram plot that took into consideration the length of every extractive summary. We selected the length of y based on the point at which the histogram plot had maximum weight. These extractive and abstractive text samples are then fed into a shared embedding layer of length z. The output of the embedding layer is an x X y X z vector for every extractive and abstractive sample.

Sentence embedding subspace

Our work aims to enhance contextual information and highlight additional aspects in our abstractive summary. This is why we developed a sentence embedding subspace that was previously not explored in previous works on summarization. To do this, we computed a similarity vector of extractive summaries (3X3) for each abstract summary sample, as illustrated in the above picture. Cosine similarity and Jaccard similarity are the similarity scores we employed in our research. Thus, for x abstractive samples, the 3D vector 3 X 3 X x is padded to n X n X x, which we convert to a 2D vector n^2 X x where y = n^2. Note y, which is the average number of words in an extractive summary, has to be the square of a natural number.

Next, to concatenate the information to get better context, a matrix addition operation is performed between the 3D vector x X y X z and the 2D vector n^2 X x. This operation ensures that the entire sentence embedding space context is included into our resultant summary. This output then constitutes the input to the next part of our model. To generate abstractive summaries, we utilized a similarity matrix-based encoder-decoder model with attention. This model takes the chosen extractive summary as input and generates an abstractive summary which we describe in next sections.

Encoder: We utilized two layers of LSTM cells in the encoder design. The embedded extractive summary vector was obtained by concatenating the three extractive summary sentences and the 3D similarity adjacency matrix created by calculating the similarity values between the three extractive summary sentences, which served as the cell's input. This ensures that both word-level and sentence-level context features are included in the input to the encoder cells.

Encoder with Attention: Neural processes involving attention [8] have been primarily studied in computational neuroscience. This concept is inspired mainly by how individuals direct their visual attention. Rather than encoding the complete source sentence in a fixed-length vector, we employ an attention method. This means the decoder can focus on different sections of the source text while producing output. Essentially, the model learns which sections to pay attention to solely based on the input sequence and previous predictions. Mathematically, at each time step (denoted as t), the model generates a context vector 'C_t' at each time step t as a weighted sum of the source hidden states as mentioned in Equation 3.

$$C_t = \sum_{t=1}^{T_x} \propto_t h_t \tag{3}$$

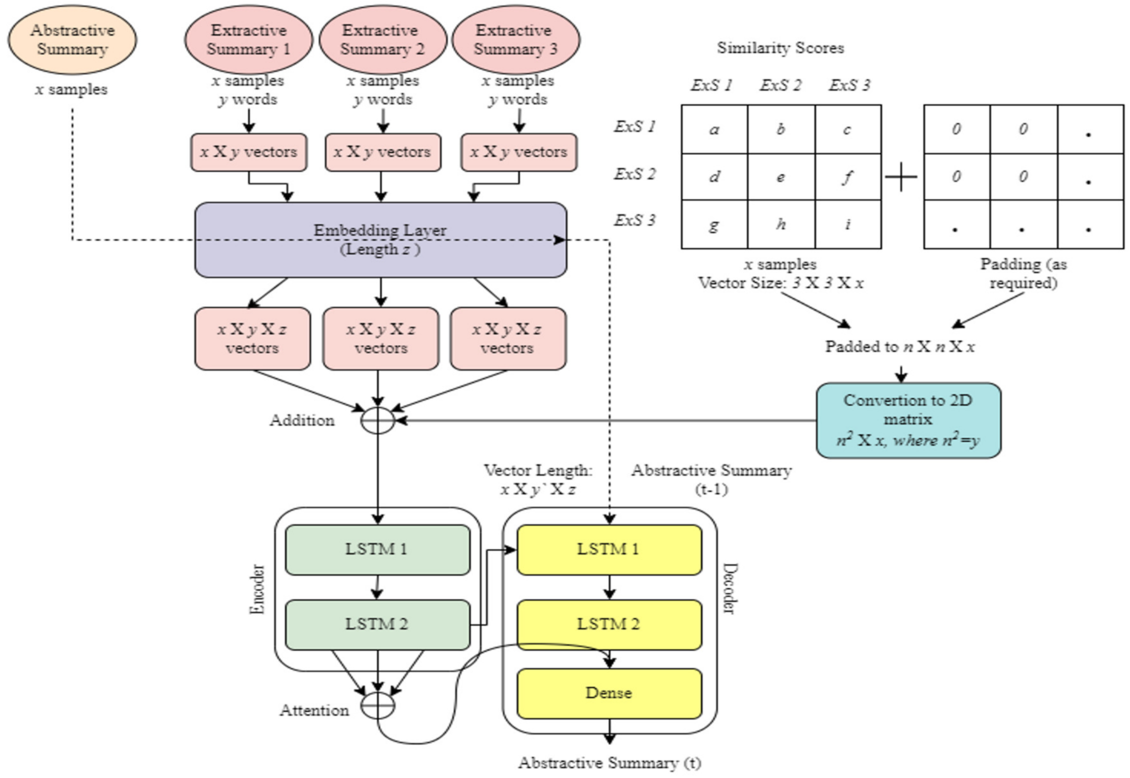

Fig.3: *SETA – Architecture.*

The attention weight ('$\propto_t$') represents the relevance of the t-th source token ('x_t') in comparison to the t-th target token ('y_t').

This is computed as:

$$\propto_t = \frac{1}{z}\exp(score(E_y(y_t-1), s_{t-1}, h_t)) \qquad (4)$$

where,

$$Z = \sum_{k=1}^{T_x}\exp(score(E_y(y_t-1), s_{t-1}, h_k)) \qquad (5)$$

In this case, Z represents the normalization constant. The function $score()$ implements a feed-forward neural network with a single hidden layer. It determines how closely the source symbol xx corresponds to the target y_t. Ey represents the target embedding lookup table, while s_t identifies the target hidden state at time t.

Decoder: The decoder uses two LSTM cells that had been initialized with the encoder's secret states. The decoder is capable of returning both sequences and states. Williams [27] established the concept of "teacher forcing" learning, as shown in this case. The input to the decoder was a one-hot tensor of abstractive summaries embedded at the word level. Meanwhile, the target data mirrored the input with a one-time step offset. The encoder passes on the initial states, which provide the information required for generation.

As a result, the decoder can generate target data for time steps beyond t, indicated by $[t+1, \ldots]$, using the input sequence and previous target predictions up to time t.

Each output time step predicts a single word, which results in the production of the complete output sequence. During model training, we use the following parameters: a batch size of 64, 100 epochs, *softmax* activation function, *rmsprop* optimizer, and $sparse\ categorical\ cross-entropy$ loss. The rate at which learning occurs was 0.001.

5. EXPERIMENTS AND RESULTS

Experiments were conducted on our custom dataset *TableSum* (section 3), evaluating the accuracy of our proposed approach using standard metrics like ROUGE-1, ROUGE-2, ROUGE-L, and BLEU. Table 4 shows the results. Furthermore, we have also conducted human evaluations on 50 random samples. Three participants were assigned the task of comparing the produced summaries against human-written summaries. They evaluated each summary using four criteria: (i) informativeness, (ii) salience, (iii) sentence coherence, and (iv) fluency and grammatical correctness.

These criterias serve specific purposes. Informativeness measures how much information the sum-

Table 4: *Automated Evaluation.*

Proposed Models	Avg. ROUGE-1	Avg. ROUGE-2	Avg. ROUGE-L	Avg. BLEU
Fine-tuned T5 Model	0.36	NA	0.31	58.2
Seq-to-Seq Model	0.21	NA	0.18	16.25
Embedding_WordVec + Similarity COSINE (SETA_v1)	0.34	0.42	0.29	38.51
Embedding_Glove + Similarity COSINE (SETA_v2)	0.31	0.36	0.27	36.7
Embedding_WordVec + Similarity Jaccard (SETA_v3)	0.32	0.34	0.21	32.21
Embedding_Glove + Similarity_Jaccard (SETA_v4)	0.26	0.36	0.13	33.23

Table 5: *Human Evaluation Results.*

Models	Results		
	Average_Fluency	Average_ Adequacy	Average_Saliency
Embedding_WordVec + Similarity COSINE (SETA_v1)	**3.91**	**4.02**	**NA**
Embedding_Glove + Similarity COSINE (SETA_v2)	**1.83**	**2.07**	**NA**
Embedding_WordVec + Similarity Jaccard (SETA_v3)	**3.61**	**3.51**	**4.02**
Embedding_Glove + Similarity_Jaccard (SETA_v4)	**2.91**	**2.02**	**2.91**
Embedding_WordVec + Similarity COSINE (SETA_v1)	**3.20**	**3.02**	**2.95**
Embedding_Glove + Similarity COSINE (SETA_v2)	**2.82**	**3.12**	**2.54**

mary delivers, salience evaluates how well the summary fits with the original content, coherence assesses sentence flow, and fluency examines the summary's grammatical quality. We assign each criterion a score ranging from 1 (poorest) to 5 (best). Table 5 shows the average scores for each criterion. We compared our results with those of previous models [28] after training them on our developed dataset *TableSum*.

As we can see in Table 4, our model SETA_v1, designed with Word2vec Embedding layer and Cosine similarity scores to prepare the similarity matrix, outperforms all the other models and is almost at par with the fine-tuned T5 model. All the different models also perform well. Therefore, choosing relevant sentences that are close to the selected summary improves the quality of the summaries. Even though the margin looks small for some parameters like Rouge-1 and ROUGE-L it is pretty substantial concerning the abstractive summary output. This is mainly because the dataset we developed is not large enough to adequately train a deep learning model. However, it is clear the quality of the summary is not affected by the size of the dataset. Table 5, which presents the human evaluation findings, shows that the model SETA_v1, which uses a word_to_vec embedding layer and cosine similarity values, consistently outperforms the current and past abstractive summary generation models [30]. Thus, our architecture can generate more informative and compact summaries, demonstrating the benefit of abstractive approaches.

6. CONCLUSION AND FUTURE SCOPE

In this paper, we address the challenging task of summarizing tables in scientific publications, an area that has received comparatively less investigation to date. We study the difficulty of table-based summarization and address the need for a proper dataset for the generation of a table summarization system. First, we identified the lack of suitable training and testing data for algorithms summarising scientific tables. To address this issue, we created the gold standard dataset known as *TableSum*, which consists of extractive and abstractive summaries of tables drawn from a wide variety of scholarly articles in different computer science fields.

Second, we discussed approaches for selecting the best extractive-abstractive summary pair for each table in scientific papers. We created a rule-based technique based on domain-specific knowledge and language principles. Our approach was validated through comprehensive evaluations, which included qualitative human assessments as well as automated indicators like ROUGE and BLEU.

Additionally, our study clarified the significance of identifying salient table summaries, a component that earlier research frequently disregarded. We addressed this by proposing SETA, a novel extractive-to-abstractive summary generation system that uses an encoder-decoder attentional technique based on sentential similarity. The outcomes of the experiments show how competitive SETA is at producing high-quality summaries. As future work, expanding our dataset to significantly improve the number and diversity of data available for table summarization presents an exciting direction for future research. This expansion could enhance the quality and efficacy of summarizing tables in scientific publications. Thus, by providing innovative solutions to present challenges and laying the framework for future advances in this field, our research has considerably advanced the field of scientific table summarising.

AUTHOR CONTRIBUTIONS

Conceptualization, M.Dey. and S.K.Mahata.; methodology, M.Dey.; software, M.Dey.; validation, M.Dey.; data curation, M.Dey., S.K.Mahata.,

A.Mondal and D.Das.; writing—original draft preparation, M.Dey, S.K.Mahata, A.Mondal.; writing—review and editing, M.Dey. and S.K.Mahata.; supervision, D.Das.; All authors have read and agreed to the published version of the manuscript.

References

[1] V. Gupta and G. S. Lehal, “A Survey of Text Summarization Extractive Techniques,” *Journal of Emerging Technologies in Web Intelligence*, vol 2, no. 3, pp. 258–268, 2010.

[2] W. S. El-Kassas, C. R. Salama, A. A. Rafea and H. K. Mohamed, “Automatic Text Summarization: A Comprehensive Survey,” *Expert Systems with Applications*, vol. 165, pp.113679, 2021.

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

[4] X. Chen, M. Li, S. Gao, R. Yan, X. Gao and X. Zhang, “Scientific paper extractive summarization enhanced by citation graphs,” *Proceedings of 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 4053–4062, 2022.

[5] S. Liu, J. Cao, R. Yang and Z. Wen, “Long Text and Multi-Table Summarization: Dataset and Method,” *arXiv preprint*, arXiv:2302.03815, 2023..

[6] C.-Y. Lin, “Rouge: A Package for Automatic Evaluation of Summaries,” *Journal of Text Summarization Branches Out*, pp. 74–81, 2004.

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

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. Polosukhin, “Attention is all you need,” *arXiv preprint*, arXiv:1706.03762v7, 2017.

[9] R. Nallapati, B. Zhou and M. Ma, “Classify or select: Neural architectures for extractive document summarization,” *arXiv preprint*, arXiv:1611.04244, 2016.

[10] C. Jianpeng, and M. Lapata, “Neural summarization by extracting sentences and words,” *arXiv preprint*, arXiv:1603.07252, 2016.

[11] Y.C. Chen, and M. Bansal, “Fast abstractive summarization with reinforce-selected sentence rewriting,” *arXiv preprint*, arXiv:1805.11080, 2018.

[12] A. Rush, S. Chopra, and J. Weston, “A neural attention model for abstractive sentence summarization,” *arXiv preprint*, arXiv:1509.00685, 2015.

[13] R. Nallapati, B. Zhou, C. Gulcehre, and B. Xiang, “Abstractive text summarization using sequence-to-sequence rnns and beyond,” *arXiv preprint*, arXiv:1602.06023, 2016.

[14] G. Jiatao, Z. Lu, H. Li, and V. OK Li, “Incorporating copying mechanism in sequence-to-sequence learning,” *arXiv preprint*, arXiv:1603.06393, 2016.

[15] C. Gulcehre, S. Ahn, R. Nallapati, B. Zhou, and Y. Bengio, “Pointing the unknown words,” *arXiv preprint*, arXiv:1603.08148, 2016.

[16] P. Liang, M. I. Jordan, and D. Klein, “Learning semantic correspondences with less supervision,” *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pp. 91-99, 2009.

[17] R. Lebret, D. Grangier and M. Auli, “Neural text generation from structured data with application to the biography domain,” *arXiv preprint*, arXiv:1603.07771, 2016.

[18] P. Ratish, and M. Lapata, “Data-to-text generation with macro planning,” *Transactions of the Association for Computational Linguistics*, vol. 9, pp.510-527, 2021.

[19] F. Wang, Z. Xu, P. Szekely, and M. Chen, “Robust (controlled) table-to-text generation with structure-aware equivariance learning,” *arXiv preprint*, arXiv:2205.03972, 2022.

[20] T. Liu, K. Wang, L. Sha, B. Chang and Z. Sui, “Table-to-text generation by structure-aware seq2seq learning,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[21] C. Li and W. Lam, “Structured Attention Networks for Table-to-Text Generation,” *ACM Transactions on Information Systems (TOIS)*, vol. 38, no. 2, pp. 1–27, 2020.

[22] Krishnamurthy, Jayant, Pradeep Dasigi, and Matt Gardner, “Neural semantic parsing with type constraints for semi-structured tables,” in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 1516-1526, 2017.

[23] L. Dong, F. Wei, and C. Tan, “Neural Text Generation in Structured Data-to-Text Applications,” in *Proceedings of (EMNLP-IJCNLP)*, pp.1321–1331, 2019.

[24] 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)*, pp.1–20, 2021.

[25] P. See, 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)*, pp. 1073–1083, 2017.

[26] R. Urbizagástegui, *Las Posibilidades de la Ley de Zipfen la IndizaciON AutomATica*, Reporte de la Universidad California Riverside, 1999.

[27] R.J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural computation*, vol. 1, no. 2 pp. 270-280, 1989.

[28] M. Dey, S.K. Mahata, and D. Das, "Exploring Summarization of Scientific Tables: Analysing Data Preparation and Extractive to Abstractive Summary Generation," *International Journal for Computers & Their Applications*, vol. 30, no. 4 , 2023.

[29] G. Sharma and D. Sharma, "Automatic text summarization methods: A comprehensive review," *SN Computer Science*, vol. 4, no. 1, 2022.

[30] S. Liu, J. Cao, R. Yang and Z. Wen, "Long text and multi-table summarization: Dataset and method," *arXiv preprint*, arXiv: 2302.03815, 2022.

Monalisa Dey is an Assistant Professor at the Institute of Engineering and Management, Kolkata (IEM) in the Computer Science and Engineering Department. She has currently submitted her Ph.D. at 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 50 research papers in reputed journals and conferences. Her research interests are Medical Data Processing, Natural Language Processing, Text Mining, and Knowledge Extraction.

Sainik Kumar Mahata is an Associate Professor and Assistant Head of Department at the Institute of Engineering and Management, Kolkata (IEM), with a focus on Computer Science and Engineering. With a Ph.D. in Engineering (CSE) from Jadavpur University, he bring over 14 years of teaching experience and 7 years of dedicated research expertise. His expertise spans Python, Natural Language Processing, and Machine Translation, showcased through 55+ research papers, 18 of which are published in esteemed international journals. In his previous roles, he was associated with JIS College of Engineering and Narula Institute of Technology as an Assistant Professor. He has completed his B.E and M.Tech, in Computer Science and Engineering, from Nagpur University and West Bengal University of Technology.

Anupam Mondal received his Ph.D. degree in 2021 from the Department of Computer Science and Engineering, Jadavpur University, India. Anupam is currently working as an Associate Professor in the Department of Computer Science and Engineering at the Institute of Engineering and Management (IEM), Kolkata, India. Before joining IEM, he worked as an Assistant Professor at S P Jain School of Global Management, Mumbai, as a Research Associate at Rolls-Royce@NTU Corporate Lab, Singapore, as a Research Assistant at Computational Intelligence Lab, NTU, Singapore, and as an Assistant Professor at JIS College of Engineering, India. His research interests are Medical Data Processing, Natural Language Processing, Text Mining, Sentiment Analysis, and Knowledge Extraction.

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.