ISSN 2007-9737

FL-T5: A Novel Hybrid Text Summarization Framework Using Fuzzy Logic and T5 Transformer

Bharti Sharma[1], Minakshi Tomer[1], Charu Gupta[2,*], Priya Dalal[1], Oscar Castillo[3]

[1] Maharaja Surajmal Institute of Technology, India

[2] Bhagwan Parshuram Institute of Technology, India

[3]Tijuana Institute Technology, Mexico

{bhartisharma, minakshi.tomer, priya}@msit.in, charu.wa1987@gmail.com, ocastillo@tectijuana.mx

Abstract. Every day, a vast amount of data on a specific topic is available on the internet, comprising both relevant and irrelevant information. Therefore, summarizing text data is crucial to comprehend the key ideas within a limited timeframe. This research introduces a novel hybrid text summarization framework called FL-T5, which combines fuzzy logic and the T5 (Transformer) model. To enhance the quality of the summarization, the FL-BART framework is employed in conjunction with extractive and abstractive summarization methods. In the extractive method, fuzzy logic applies fuzzy rules to create sentence embeddings, while the abstractive summarization method utilizes Sequence-to-sequence recurrent neural networks and Transfer Learning with a Unified Text-to-Text Transformer approach. Each sentence is scored based on relevance and novelty metrics to generate an intermediate summary. The sentences with the highest scores are selected for the summary, which serves as input to the abstractive method. Extensive experiments were conducted to evaluate the effectiveness of FL-T5. The benchmark datasets used for abstractive summarizationincludeCNN/DailyMailandDUC200while DUC-2006 and DUC-2007 are employed for extractive summarization. The recall-oriented understudy for gisting evaluation (Rouge) metric is used for evaluation. The results indicate that the proposed FL-T5 framework excels in efficient text summarization and outperforms other state-of-the-art models in terms of generated results.

Keywords. Abstracrive summarization, fuzzy logic, text summarization, CNN rules.

1 Introduction

Along with the significant advancement in technologies, a vast amount of information and documents available on the internet. Human summarization is subject to bias, is context-dependent, and may vary according to individual cognition.

Thus, appropriate methodologies and tools are required to extract pertinent and imperative portions in order to obtain critical information in the form of a summary, resulting in a machine-generated summary that is devoid of bias.

This sparks many researchers to develop a technical solution capable of automatically summarizing texts.

Automatic text summarization generates summaries that include all pertinent information from the original document [1]. Thus, information is delivered swiftly while maintaining the document's original intent [2]. Researcher and academician focused on the text summarization in the mid of 20th century but it was first described publically in [3] using statistical technology named as word frequency diagram.

Text Summarization (TS) is the procedure of generating a condensed version of text without compromising the crucial information and meaning of the text. Since manual text summarization is a period costly and, for the most part, a relentless

task, the automation of the study is acquiring expanding prominence and comprises a solid inspiration for scholarly exploration. Text summarizers have been shown to be useful in a variety of application domains, ranging from stock market forecasting to keyword extraction for search engine optimization. [4,5] have proposed automatic Email summary creation utilizing cue phrases.

Automatic text summarization are primarily focused on summarizing single document like research article , news article , weather forecast etc. and multi documents including customer review , multiple source –news , article , emails etc. [6,7]. Abstractive and extractive text summarizing are two separate types of TS based on the summary results. Summaries can also be classified as generic and query-focused summaries.

As the name suggests, query-focused summaries [3] contain the content related to the query, whereas generic summaries contain a general summary of the information present in the document [8]. Based on output, summaries can be of two types: indicative and informative outlines. Indicative summaries provide us the topics covered in the document and the idea of the content. Whereas, Informative summaries along with the issues provide a detailed explanation for each topic.

Deep learning algorithms beat other approaches when it comes to text summarization. Also, there have been some hybrid approaches with fuzzy that have improved the results. In this paper, we proposed a hybrid model which consists of a Fuzzy algorithm and T5-Text-to-text Transformer. Using both extractive and abstractive TS improves results. Abstractive TS uses T5 transformer, and extractive TS uses fuzzy logic. The T5 model is an incredibly powerful abstractive text summarize, and along with fuzzy logic. Compared to the old results, the new outcomes have been far better.

The remaining text of the article is constructed as follows: In Part 2, we give a literature review of text summary techniques. The objective of this research is discussed in section 3, which follows. In Section 4, the background of the hybrid strategy is presented. In Section 5, the data set and experimental outcomes are presented. Section 6 of the study presents some potential future research directions.

2 Literature Survey

In 2015, [9] were the first to employ deep learning for abstractive text summarization. Convolutional encoders, bag-of-words encoders, and attention-based encoders were all used in the model given by Rush et al. The input is used to condition the words that make up the summary, which is constructed using a local attention model. In addition, the decoder conducted a beam search to determine which summary terms were most appropriate. We trained on the Gigaword dataset, and then utilized DUC-2003 and DUC-2004 to test and refine our abstract. The RNN encoder-decoder model with an attention mechanism became the standard for abstractive text summarization following the RAS study [10,11]. Encoder and decoder hidden state sizes were matched [12]. RNN encoder-decoder model's attention mechanism. Word-level and sentence-level bidirectional GRU-RNN encoder layers make up the architecture [13]. Nonetheless, a unidirectional GRU-RNN was employed in the decoder [10]. Training was done with the use of the Gigaword, DUC, and CNN/Daily datasets. Tokenization, entity recognition, and part-of-speech tagging were just some of the preprocessing steps applied to the datasets. An attention mechanism was proposed name as selective encoding for abstractive sentence summarization model(SEASS) using a bidirectional GRU [14].This model has three parts : an encoder , a selective gate network and a decoder. This model has three parts: an encoder, a selective gate network, and a decoder. While a unidirectional GRU makes up both the encoder and the decoder, a bidirectional GRU is used in the former. Yet the selected gate is what actually produces the words that make up the representation of the sentences. An abstract text summary approach, called the dual attention sequence-to-sequence model, was suggested in [9]. Two generalized recurrent unit (GRU) encoders work in tandem with a gate network to form the dual attention decoder in this model [15]. Studies prior to this one used abstractive text summarization to produce a single sentence. In

contrast, the models were used in the following studies, although only in elaborate summaries. A single layer encoder decoder model was suggested in [16] to generate multi-sentence abstract using separate sequence to sequence attention mechanism. The bidirectional LSTM serves as the encoder, while the unidirectional LSTM serves as the decoder. Tests were conducted using data from CNN and the Daily Mail. The words were transformed into vectors using pretrained word embedding. To construct an abstractive summary, the adversarial framework was developed [17]. The resulting summary was initially optimized using reinforcement learning. Then, a discriminator was implemented to determine if the output summary was machine-generated or based on the underlying ground truth. An LSTM-CNN model was suggested in [38] for ATSDL based on exploration of phrases-level semantics. CNN and Daily/Mail datasets were used during model training. In[18], an abstractive and extractive generative model was developed as a means of steering the generation process. [11] architecture served as the basis for the reference generation. An abstractive text summarization approach was discussed in [40], where the decoder was decomposed into a pertained language model was suggested [19]. This model consist of single layer bidirectional LSTM encoder with three layer of unidirectional weight dropped LSTM. In addition, CNN/Daily Mail datasets were used throughout the model's development and evaluation. Both qualitative and quantitative methods of assessment were used. One hundred full texts were chosen at random, and five evaluators assessed them for readability and relevance as part of the qualitative evaluation. Nevertheless, ROUGE1, ROUGE2, and ROUGE-L were used in the quantitative analysis.in contrast, model using two layer encoding and single layer decoding GRU unit in [50]named as dual encoding paradigm for abstractive text summarization(DEATS)[20]. All of the tests were run using the CNN/Daily Mail and DUC 2004 datasets. Model proposed [21-23] were based on BERT word embedding techniques. Both the Wang et al. and Egonmwan et al. models use a bidirectional GRU at the encoder and a unidirectional GRU at the decoder for their respective techniques. Each model was tested with the CNN/Daily mail datasets. A double attention pointer network (DAPT) was proposed [24], which consisted of an LSTM encoder and an LSTM decoder. For the purpose of abstract summarization, we integrated semantic data transformation with deep learning approaches [25]. Encoder-decoder deep learning model was used to provide summary statement. When it was done, the resulting summary was formatted for human consumption.

Many successful neural models have utilized an encode-decoder structure [12]. BERT is an emerging model which is trained on masked language [26]. Sequence to Sequence summarizers can be equipped with reinforcement learning [27], but it is not guaranteed that generated summaries convey the same meaning as the input document. Reinforcement learning [28] has also been applied to perform text summarization.

Fuzzy logic has also been successfully applied in the field of text summarization [29]. A graph algorithm is presented based on a fuzzy set for analyzing text documents in [25], demonstrating that extracting specific topics is more informative than extracting sentences or keywords. Fuzzy summarization techniques have been compared with decision-tree and naive Bayes algorithms (machine learning). Neural network models do well in natural language processing tasks, which improves their ability to summarize abstract concepts. Hybrid text summarization models of deep learning algorithms and fuzzy logic are proposed [30], giving better precision, recall, and F-measure results. A novel summarization model was proposed in [31] based on fuzzy logic and sequence to sequence model. An attention mechanism was also introduced, which made the mode noise immune. When combined with neural network-based models in extractive Summarization, Fuzzy logic produced promising results [31].

3 Objectives

The goal is to propose a framework that combines extractive and abstractive summarization methods to overcome their individual limitations. In the extractive phase, the framework selects crucial

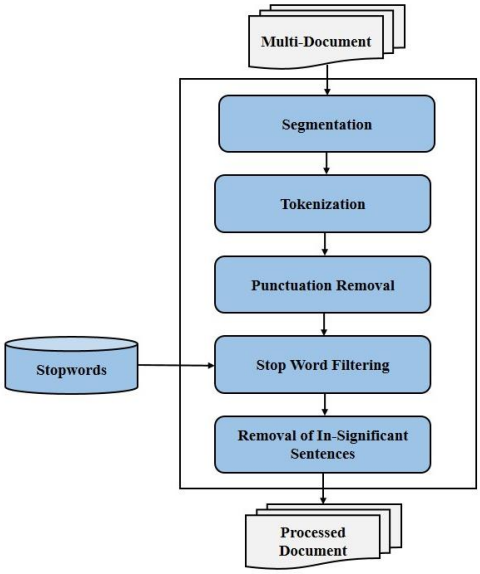

Fig. 1. Text Preprocessing Steps

sentences, reducing the dataset size and eliminating redundancy, in preparation for the abstractive phase. During the abstractive phase, the selected sentences from the extractive summarization are rephrased using newly generated words, resulting in a summary that closely resembles human-written text. By providing shorter, yet higher-quality input sentences, the use of extractive summaries also leads to shorter training times.

4 Proposed Model fuzzy+T5 for TS

In this section, an Extractive-Abstractive hybrid model for text summarization is proposed. The proposed model incorporates features of both abstractive and extractive methods for TS. Fuzzy logic, accomplished using fuzzy logic, uses fuzzy inference rules to minimize the length of input text by picking more informative sentences and help add context to the document. Extractive output is then advanced through the T5 model to produce the resultant abstractive summary.

The proposed strategy entails the following four steps (i) the text is preprocessed, (ii) features are extracted, (iii) Extractive summarized using fuzzy logic, and (iv) Abstractive Summarization with T5 Transformer.

4.1 Phase 1: Text Preprocessing

Preprocessing is the commencing and crucial process for transforming unstructured text data to structured data that can be further used to generate output summaries. Text Preprocessing consists of several steps performed sequentially, and different sequences of the same steps can produce different results. It incorporates splitting the text into multiple sentences, segmenting them into words (tokens), removing special characters of no meaning, and doing away with stop words.

4.1.1 Segmentation and Tokenization

The initial process of preprocessing consists of segmentation and tokenization. It begins with identifying different components of a sentence and then segmenting them. The next step is to tokenize your pieces of text into their words/tokens, which are used to create so-called vocabularies that will be used in the language model you plan to build as shown in Figure 1.

4.1.2 Removing Stop Words and Stemming

After performing tokenization, all the words that contribute nothing to a sentence's meaning are removed(adverbs, pronouns, prepositions, articles). Stop-word filtering is implemented to reduce dimensionality in the language of the text. After removing stop words, we perform stemming, reducing the inflected or derived word into its base form.

4.2 Phase 2: Feature Extraction

After preprocessing of data, feature extraction is the following process. Features or characteristics of data are the factors that convey information about the given text that any statistical technique or machine learning/mathematical model can understand. Features of a dataset are expressed in the form of vectors or numerics. In-Text Summarization, features are the indicators of meaning or type of text used to produce a summary. The most used features are Sentence Length, Sentence Position, Title Words, Tf-IDF, Keywords, Thematic Words, Proper Nouns,

Cosine-similarity, Numerical Data, Bigrams, Trigrams, N-grams, Semantic Terms, And Frequent Semantic, etc.

Out of all the features, as mentioned earlier, Nine (9) features, including Sentence Position, Sentence length, Bigrams, Trigrams, TF-IDF, Proper Noun Score, Cosine Similarity, and Numeric Token were selected and used to produce output summary.

4.2.1 Sentence Position

Paragraph placement can affect how a sentence reads in context, whether it's within a larger body of text or not. Generally speaking, the first and last sentences of a paragraph have more weight. In order to compute the sentence position score, we must take into account both the total number of sentences as well as the index of the sentence using the equation(1) that has been provided:

$$SP = \frac{Total\ No.of\ Sentences - IS}{Total\ No.of\ Sentences}, \quad (1)$$

where SP=Sentence Position, IS=Index of Sentence.

4.2.2 Bigrams

Words that are next to one other in the input string can be in any order, making them a Bigram. It is a tool for counting pairs of adjacent words in a document.

4.2.3 Trigrams

The term "trigram" refers to any input string sequence that has three words that are next to one another. It is used to determine the total number of sets of three distinct adjacent words that are contained within the document.

4.2.4 TF-IDF

The strategy relies on statistical methods and syntactic in the goal of determining the most important words in a document in order to convey its meaning.TF is used to figure out the relative frequency of words in a given sentence in its topic description. Equations are used to figure out TF-IDF. TF is relative frequency of word appearing in a document, IDF is the inverse document frequency of the word across a set of documents. TF can emphasize the importance of terms used frequently in a document. TF-IDF is calculated using equations (2-4):

$$TF(t) = \frac{No.\ of\ Times\ term\ t\ appear\ in\ the\ document}{Total\ Number\ of\ terms\ in\ the\ document}, \quad (2)$$

$$IDF(t) = log\frac{Total\ No.of\ Documents}{No.of\ Documents\ containing\ the\ term\ t}, \quad (3)$$

$$TF - IDF(t) = TF(t) \times IDF(t). \quad (4)$$

4.2.5 Cosine-Similarity

Cosine similarity is a measure between two non-zero vectors of an inner product space. The vectors are then normalized so that their length is 1, and the cosine of the angle between them equals the inner product of the normalized vectors. For the given text, the centroid of the sentence vector is computed, and further cosine similarity is utilized to calculate the similar amplitude of the sentence with the centroid.

4.2.6 Thematic Score

These are highly relative domain-specific words. Each feature's score is calculated as the ratio of its minimum and maximum values, where minimum and maximum values are the number of thematic words that occur in a phrase.

4.2.7 Sentence Length

Sentence length can emphasize the importance of the sentence. It can be observed that longer sentences can have a higher probability of containing the context of the document given. The length of the sentence actually represents the number of tokens (words) present in it. This process measures sentence length based on the average sentence length in the document instead of using absolute length. Eq. (5):

$$SL(L) = \frac{Length\ of\ Sentence}{Length\ of\ Longest\ Sentence}, \quad (5)$$

where SL= Sentence Length.

4.2.8 Proper Noun Score

Unless they are accompanied by the appropriate noun, pronouns are not considered significant features and are therefore omitted from the list. It can be calculated as a ratio. of the number of

Algorithm 1: Proposed Hybrid Model Based on FL-T5

Input : Reference Summary
Output: System Generated Summary.
Step 1: The Preprocessing is done on the input document using (Tokenization, Segmentation etc.)
Step 2: Construct a table for word embedding using pre-trained work vectors
Step 3: Calculate the Feature count using Nine-Feature.
CS = Cosine Similarity
TS = Thematic Score
SP = Sentence Position
TFD = TF-IDF
PN = Proper Noun Score
NT = Numeric Token
BS = Bigrams
TS = Trigrams
SL = Sentence Length
Step 4 : Consequent Object hold universe variable(CS,TS,SP,TFD,SL,PN,NT,BS,TS)
Step 5 : Fuzzy Control System like Fuzzy variable using a set of rules and sentences are
Scored using the rules.
Step 6: J = Fuzzy system generate the Extractive Output
Step 7: K = Extractive output is fed as input to FL-T5 model
Step 8: Final output summary is generated
Step 9: Final output summary is used to calculate Rouge between References Summaries

appropriate nouns present in the sentence to the number of tokens (length).

Proper Noun Score (SP) can be seen in the following Eq. (6):

$$PNS\ (N) = \frac{Total\ no.of\ PN\ in\ sentence}{Total\ no.of\ words\ in\ sentence}, \quad (6)$$

where PNS= Proper Noun Score.

4.2.9 Numeric Token

It means how many words are in the sentence. Each sentence or section of text is broken down into smaller meaningful pieces called tokens. Words, sentences, sub words (such as n-grams), or even individual characters could make up their make-up. The value of its score is determined by the following formula [12]:

$$NT(S) = \frac{Total\ no.of\ numeric\ data\ in\ sentence\ S}{Total\ no.of\ words\ present\ in\ sentence\ S}, \quad (7)$$

where NT= Numeric Token.

4.3 Fuzzy Rules for TS

A fuzzy-based system is utilized to generate the extractive summary of the text. In this system, a feature matrix is used to extract good sentences from the text and a triangular membership function is used to calculate the degree of truth of a sentence.

A sentence is mainly divided into two parts: antecedent and consequent. By using the 'g' triangular membership function, the antecedent is further divided into: Poor, Average and Good and the consequent is divided into: Unimportant, Average and Important sentences. Seven sets of rules are constructed by using "IF-THEN" rules to categorize the sentences, as shown in Table 1.

The Mamdani inference model is used to process these rules and generate a fuzzified output. This output is then defuzzified to a crisp value and final scores for each sentence is acquired[32]. A mean membership function is calculated using the centroid method. The sentences with high scores are considered to form a summary of the text document which is then passed as the input to the T5 Transformer for abstractive text summarization.

The high-ranking sentences from the list of all sentences are then ordered by frequency in the original document and added to the original document to create the final summary.

4.4 T5 Transformer for TS

T5 model that is used in proposed work based on transfer learning. T5's superior generalization over BERT's discussed in [33].T5 can be used for providing responses to simple inquires other than just translation.

Transformer are the backbone of T5. Transformers have significantly impacted the field of text summarization, particularly abstractive summarization. Transformers are built on the foundation of attention mechanisms. This enables them to understand the relationships between words in a text and capture context effectively. This is crucial for summarization, as the model needs to identify the important information to include in the summary.

Table 1. Fuzzy Rules for extractive text summarization

IF	THEN
Proper Noun(PN) = GOOD Numeric Token(NT)= GOOD TF-IDF score(IFD) = GOOD Sentence Position(SP) = GOOD	Sentence = Important
Sentence position(SP) = GOOD Numeric token (NT) = AVERAGE	Sentence = Important
Thematic score(TS) = GOOD Numeric Token (NT)= GOOD	Sentence = Important
Cosine-similarity(CS)= GOOD Sentence Position(SP) = GOOD	Sentence = Average
Bigram(BS) = GOOD Trigram(TS)= GOOD Proper noun(PN) = AVERAGE Numeric token (NT)= AVERAGE	Sentence = Unimportant
Proper noun (PN) = POOR Thematic score(TS) = POOR	Sentence = Unimportant
Proper Noun(PN) = POOR IF-IDF (IFD)= AVERAGE Sentence Position(SP) = POOR Numeric Token (NT) = POOR Proper Noun (PN) = POOR	Sentence = Unimportant

Each transformer has 2 distinct layer one is encoder layer and another is decoder layer. The encoder processes the input text and converts it into a compressed representation, capturing its essence. The decoder then generates the summary based on this representation.

The self-attention mechanism in Transformers allows the model to weigh the importance of different words in the input text when generating the summary.

Transformers excel at capturing contextual information, allowing them to generate coherent and contextually accurate summaries. They understand not just individual words, but also the relationships between them, leading to better summaries.

Pre-trained Transformers (such as BERT or GPT) are often fine-tuned on summarization-specific datasets. This fine-tuning adapts the model to the summarization task, making it better at generating concise and coherent summaries. Transformer allow the model to copy words or phrases directly from the source text to the summary, which is particularly useful for maintaining factual accuracy. Coverage Mechanisms used to address the issue of missing out on important content, coverage mechanisms are used. The choice of model, training data, fine-tuning strategy, and decoding techniques all play crucial roles in achieving high-quality summaries.

The information flow between the transformer's six encoding devices and six decoding devices is seen in Figure 2. Each encoder is made up of two layers: a position-wise completely connected feed-forward network and a multi-head self-attention layer. A decoder typically consists of a masked multi-head self-attention layer [26] in addition to the two layers already mentioned.

5 Experimental Setup and Results

To assess the efficiency of the suggested multi-document summarization approach, this section outlines the experimental environment. All evaluations and modifications were conducted using a Windows system equipped with an Intel CPU running at 2.50 GHz and 4GB of RAM.

5.1 Dataset

Various types of common datasets can be used to compile summaries. The CNN/Daily Mail dataset [34], which is used for the experiments, is a good starting point for summarizing text in general. There is a multi-sentence summary of the news story for your use. When the CNN and Daily Mail datasets are put together, there are more training examples.

The merged dataset is trained, tested, and analyzed with different sets. During the training, testing, and validation phases, a total of 311 672 articles were used. Of those, 249,337 were used for training. The DUC-2002 and DUC-2004 standards made by NIST are also used. The quality of the summary is judged by both the

Table 2. Characteristics of datasets

Dataset	Domain	Document count	Summary Limit
DUC 2004	News	500	100 words
DUC2006	News	1300	250 words
DUC 2007	News	1200	250 words
CNN/Daily Mail	News	2,87,000	-

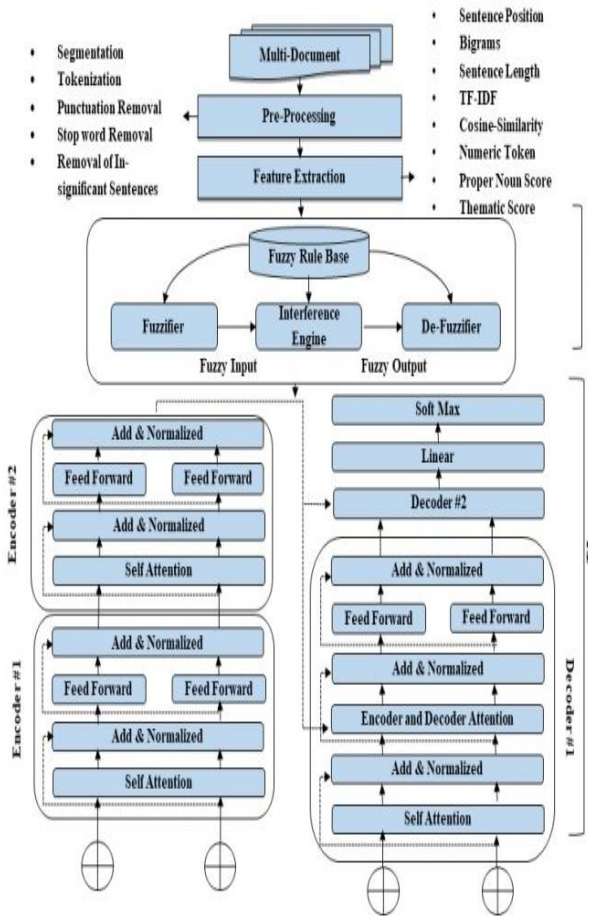

Fig. 2. Proposed technique FL-T5 based on Text Summarization

corpus and the test data. Table 2 gives information about the text that summarizes the datasets that can be used for research. The summaries are analyzed via ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [35], which compares them to a "gold standard" summary by counting how many words are used in both.

5.2 Experimental Setup

This section presents the dataset used followed by the system setting applied for performing the experimentations along with the metric used for evaluation to construct a framework. State-of-the-art models are used to evaluate the FLT5's output. CNN/Daily Mail is used as a training and a validation set for the model. Embeddings of different sizes can be generated by retraining the model. While the hidden state dimension is 256, the word embedding dimension is 128. By default, the software use a lexicon of 50,000 words. The learning rate is set to 0.1. The network's parameters are initially seeded randomly within the range [0.05, 0.05]. The Adagrad algorithm is also used in this process. The model can handle OOV terms on its own, which contributes to its adaptability to smaller vocab sizes.

5.3 Metrics for Evaluation

Both intrinsic and extrinsic methods can be used to evaluate the quality of a summary of text. To that end, an evaluation of the intrinsic type is being carried out here. Standards-tested the ROUGE toolbox figure out how many n-grams the model-made summary and the reference (best) summary have in common. Its purpose is to rate abstractions of texts. It has metrics that can be used to instantly determine a summary's quality in comparison to exemplars crafted by people. ROUGE-N calculates the fraction of N-grams shared between a model's generated summary and an ideal summary.

ROUGE-L is capable of calculating the LCS (Longest Common Subsequence) metric. The accuracy, F-measure and recall are calculated by Rouge:

$$\text{Precision} = \frac{\text{Reference summary} \cap \text{System generated summary}}{\text{System generated summary}}, \quad (8)$$

$$\text{Recall} = \frac{\text{Reference summary} \cap \text{System generated summary}}{\text{Reference summary}}. \quad (9)$$

The following gives a summary of F-measure:

$$F\text{-measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (10)$$

6 Results

In this section, we assess the proposed FL-T5 framework in contrast to state-of-the-art models using the specified datasets.

CNN/Daily Mail Dataset: The CNN/Daily Mail Dataset has been a focal point for abstractive text summarization, addressing critical challenges in the field. Attentional Encoder Decoder [9] was introduced as a solution, focusing on aspects like keyword modeling and sentence hierarchy. Similarly, SummaRunner-abs [36] and SummaRuNNer aimed to capture sentence structure hierarchy, thereby enhancing performance. Other abstractive techniques [10] also tackled fundamental issues, including keyword modeling and structural hierarchy, to improve summarization quality.

ABS and ABS+ [30], utilizing a standard encoder-decoder with attention mechanisms, were applied to the CNN/Daily Mail dataset. An alternative choice, seq2seq+atten [37], leveraged sequence-to-sequence models with attention. PGC [16] introduced a sequence-to-sequence model with coverage to handle out-of-vocabulary words, while SummaRuNNer-abs [36] transformed from an extractive model to an RNN-based approach.

The Hybrid Approach to Simulation (HATS) [38] was developed to simulate human learning processes in a learning paradigm. Additionally, FLSTM [29], an ensemble model combining fuzzy and LSTM approaches, was compared to it.

In the context of DUC-2004, various topic detection methods were employed, including TOPIARY [48], ABS and ABS+[9], which introduced the first machine translation model. RAS-Elman [11] adopted an encoding and decoding approach, featuring an attention-based encoder and a recurrent neural network-based decoder. Additionally, attention-based techniques like wordslvt5k-lsent [10] and SEASS [14], incorporating a selective gate network in its encoder-decoder, were utilized. For extractive

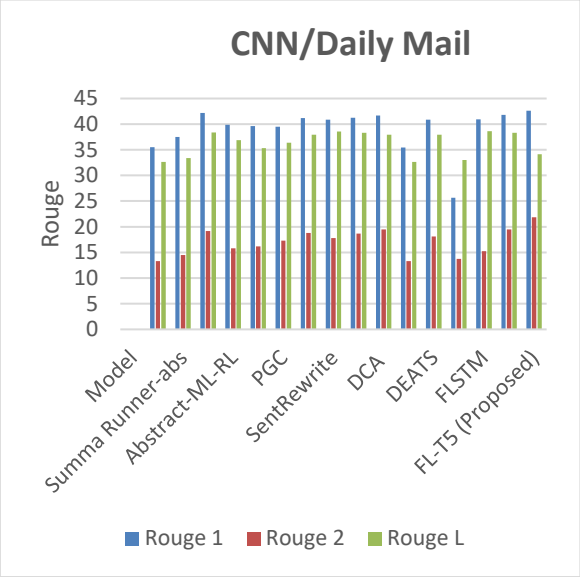

Fig. 3. Comparing the performance of FL-T5 against the latest models on the CNN/Daily Mail dataset.

Table 3. Assessing the performance of existing models using the CNN/DM dataset

Model	*Rouge 1*	*Rouge 2*	*Rouge L*
Words-lvt2k	35.5	13.30	32.65
Summa Runner-abs	37.5	14.50	33.40
HATS	42.16	19.17	38.35
Abstract-ML-RL	39.87	15.82	36.9
SummaRuNNer	39.6	16.2	35.3
PGC	39.53	17.28	36.38
SummCoder	41.20	18.78	37.96
SentRewrite	40.88	17.8	38.54
Bottom-Up Summ	41.22	18.68	38.34
DCA	41.69	19.47	37.92
ABS	35.46	13.30	32.65
DEATS	40.85	18.08	37.96
ABS+	25.63	13.75	33.01
FLSTM	40.96	15.22	38.63
BERT	41.82	19.48	38.30
FL-T5 (Proposed)	**42.62**	**21.85**	**34.12**

Table 4. Assessing the performance of existing models using the DUC2004 dataset

Method	Rouge 1	Rouge 2	Rouge L
LSA	29.73	3.99	26.34
RAS-ELMAN	28.97	8.26	24.06
ABS+	28.18	8.49	23.81
SummCoder	29.31	9.54	25.24
LexRank	31.21	5.25	25.01
SEASS	29.21	9.56	25.51
RBM	25.46	8.27	42.38
Words-lvt5k-lsent	28.61	9.42	25.24
ABS	26.55	7.06	22.05
DEATS	29.91	9.61	25.95
TOPIARY	25.12	6.46	20.12
FL-T5 (Proposed)	43.57	30.10	34.50

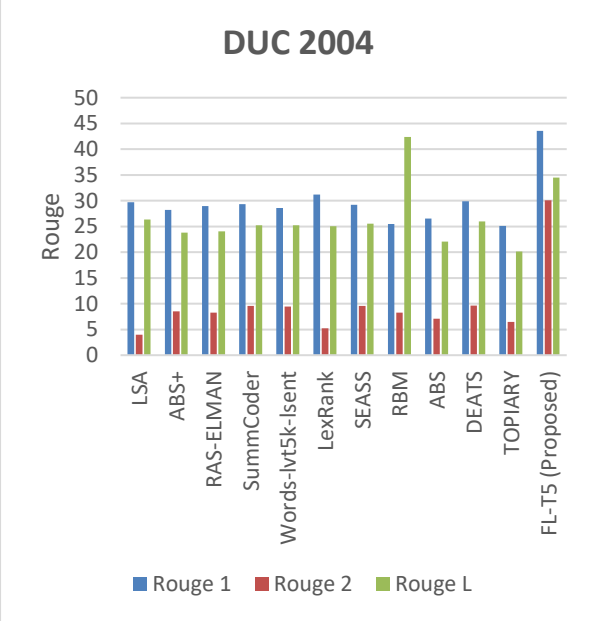

Fig. 4. Comparing the performance of FL-T5 against the latest models on the DUC2004 Mail dataset

summarization of texts, methods such as SummCoder [12]and DEATS [39] were applied.

Table 3 and Figure 3 showcase results from FL-BART experiments conducted on the CNN/Daily Mail dataset. This model exhibits exceptional performance on the CNN/Daily Mail Rouge Dataset, surpassing its competitors. Specifically, it achieves a Rouge-1 score of 47.41, a Rouge-2 score of 24.57, and a Rouge-L score of 38.04, demonstrating superior performance compared to other methods. Table 4 and Figure 4 present the evaluation results for DUC-2004, displaying high Rouge scores with a Rouge-1 count of 40.98, Rouge-2 count of 33.57, and Rouge-L count of 38.43. The model also performs well on the DUC-2006 dataset, with favorable Rouge scores.

As shown in Table 5 and Figure 5, it achieves a Rouge-1 count of 39.04, Rouge-2 count of 15.16, and Rouge-L count of 20.65 on DUC-2006, and Rouge-1 count of 46.09, Rouge-2 count of 21.80, and Rouge-L count of 25.43 on DUC-2007. Notably, the CNN/Daily Mail dataset yields higher Rouge scores overall due to its larger size, making it a promising choice for model training. Qualitative testing across multiple datasets further validates its utility.

7 Conclusion

This research introduces FL-T5, a novel framework for automatic TS that combines extractive and abstractive approaches. The framework enhances the semantic representation of sentences by generating extractive summaries using Fuzzy Rules. These extractive summaries are then used as input to the T5 model, which generates the abstractive summary. The combination of these approaches improves the efficiency of both the extractive and abstractive models. Results on the CNN/Daily Mail dataset demonstrate that this technique outperforms state-of-the-art models according to the Rouge metric. Additionally, when tested on the DUC-2004 datasets, it achieves higher ROUGE-1 and ROUGE-2 scores. The ROUGE-L score is comparable to that of other models. The proposed hybrid model is also evaluated on the DUC-2006 and DUC-2007 datasets, which are commonly used for extractive summarization, showing promising results. In future work, the F1 count could be improved by incorporating a different extractive model trained

with sentence embedding. Furthermore, combining additional abstractive models such as reinforcement learning, transformers, and hybrid training functions could further enhance the performance.

References

1 **Gambhir, M., Gupta, V. (2017).** Recent Automatic Text Summarization Techniques: Survey. Artificial Intelligence Review, Vol. 47, No. 1, pp. 1–66. doi: 10.1007/s10462-016-9475-9.

2 **Khan, A. et al. (2018).** Abstractive Text Summarization Based on Improved Semantic Graph Approach. International Journal of Parallel Programming, Vol. 46, No. 5, pp. 992–1016. doi: 10.1007/s10766-018-0560-3.

3 **Luhn, H.P. (1958).** The Automatic Creation of Literature Abstracts. IBM Journal of Research and Development, Vol. 2, No. 2, pp. 159–165. doi: 10.1147/RD.22.0159.

4 **Gerani, S., Mehdad, Y., Carenini, G., Ng, R.T., Nejat, B. (2014).** Abstractive Summarization of Product Reviews Using Discourse Structure. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1602–1613. doi: 10.3115/v1/D14-1168.

5 **Schilder, F. et al. (2017).** A Study of Abstractive Summarization Using Semantic Representations and Discourse Level Information. In Text, Speech, and Dialogue, pp. 482–490. doi: 10.1007/978-3-319-64206-2_54.

6 **Zajic, D.M., Dorr, B.J., Lin, J. (2008).** Single-Document and Multi-Document Summarization Techniques for Email Threads Using Sentence Compression. Information Processing & Management, Vol. 44, No. 4, pp. 1600–1610. doi: 10.1016/j.ipm.2007.09.007.

7 **Patel, V., Tabrizi, N. (2022).** An Automatic Text Summarization: Systematic Review. Computación y Sistemas, Vol. 26, No. 3, pp. 1259–1267. doi: 10.13053/CyS-26-3-4347.

8 **Neri-Mendoza, V., Ledeneva, Y., García-Hernández, R.A., Hernández-Castañeda, Á. (2024).** Multi-Document Text Summarization through Features Relevance Calculation. Computación y Sistemas, Vol. 28, No. 3, pp. 1417–1427. doi: 10.13053/CyS-28-3-5201.

9 **Rush, A.M., Chopra, S., Weston, J. (2015).** A Neural Attention Model for Abstractive Sentence Summarization. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 379–389. doi: 10.18653/v1/D15-1044.

10 **Nallapati, R., Zhou, B., Gulcehre, C., Xiang, B. (2016).** Abstractive Text Summarization Using Sequence-to-Sequence RNNs and Beyond. Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, pp. 280–290. doi: 10.18653/v1/K16-1028.

11 **Chopra, S., Auli, M., Rush, A.M. (2016).** Abstractive Sentence Summarization with Attentive Recurrent Neural Networks. Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 93–98. doi: 10.18653/v1/N16-1012.

12 **Joshi, A., Fidalgo, E., Alegre, E., Fernández-Robles, L. (2019).** SummCoder: Unsupervised Framework for Extractive Text Summarization Based on Deep Auto-Encoders. Expert Systems with Applications, Vol. 129, pp. 200–215. doi: 10.1016/j.eswa.2019.03.045.

13 **Cheng, J., Lapata, M. (2016).** Neural Summarization by Extracting Sentences and Words. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pp. 484–494. doi: 10.18653/v1/P16-1046.

14 **Zhou, Q., Yang, N., Wei, F., Zhou, M. (2017).** Selective Encoding for Abstractive Sentence Summarization. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pp. 1095–1104. doi: 10.18653/v1/P17-1101.

15 **Cao, Z., Li, W., Wei, F., Li, S. (2018).** Retrieve, Rerank and Rewrite: Soft Template Based Neural Summarization. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, pp. 152–161. doi: 10.18653/v1/P18-1015.

16 **See, A., Liu, P.J., Manning, C.D. (2017).** Get to the Point: Summarization with Pointer-Generator Networks. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pp. 1073–1083. doi: 10.18653/v1/P17-1099.

17 **Liu, L., Lu, Y., Yang, M., Qu, Q., Zhu, J., Li, H. (2018).** Generative Adversarial Network for Abstractive Text Summarization. Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 32, No. 1. doi: 10.1609/AAAI.V32I1.12141.

18 **Neri-Mendoza, V., Ledeneva, Y., García-Hernández, R.A., Hernández-Castañeda, Á. (2023).** Generic and Update Multi-Document Text Summarization Based on Genetic Algorithm. Computación y Sistemas, Vol. 27, No. 1, pp. 269–279. doi: 10.13053/CyS-27-1-4538.

19 **Kryściński, W., Paulus, R., Xiong, C., Socher, R. (2018).** Improving Abstraction in Text Summarization. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 1808–1817. doi: 10.18653/v1/D18-1207.

20 **Yao, K., Zhang, L., Du, D., Luo, T., Tao, L., Wu, Y. (2020).** Dual Encoding for Abstractive Text Summarization. IEEE Transactions on Cybernetics, Vol. 50, No. 3, pp. 985–996. doi: 10.1109/TCYB.2018.2876317.

21 **Wang, Q., Liu, P., Zhu, Z., Yin, H., Zhang, Q., Zhang, L. (2019).** A Text Abstraction Summary Model Based on BERT Word Embedding and Reinforcement Learning. Applied Sciences, Vol. 9, No. 21, pp. 4701. doi: 10.3390/app9214701.

22 **Liu, Y., Lapata, M. (2019).** Text Summarization with Pretrained Encoders. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pp. 3730–3740. doi: 10.18653/v1/D19-1387.

23 **Egonmwan, E., Chali, Y. (2019).** Transformer-Based Model for Single Documents Neural Summarization. Proceedings of the 3rd Workshop on Neural Generation and Translation, pp. 70–79. doi: 10.18653/v1/D19-5607.

24 **Li, Z., Peng, Z., Tang, S., Zhang, C., Ma, H. (2020).** Text Summarization Method Based on Double Attention Pointer Network. IEEE Access, Vol. 8, pp. 11279–11288. doi: 10.1109/ACCESS.2020.2965575.

25 **Kouris, P., Alexandridis, G., Stafylopatis, A. (2022).** Abstractive Text Summarization Based on Deep Learning and Semantic Content Generalization.

26 **Devlin, J., Chang, M.-W., Lee, K., Toutanova, K. (2018).** BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv:1810.04805. doi: 10.48550/arXiv.1810.04805.

27 **Sutskever, I., Vinyals, O., Le, Q.V. (2014).** Sequence to Sequence Learning with Neural Networks. Advances in Neural Information Processing Systems, Vol. 27.

28 **Paulus, R., Xiong, C., Socher, R. (2017).** A Deep Reinforced Model for Abstractive Summarization. arXiv preprint arXiv:1705.04304. doi: 10.48550/arXiv.1705.04304.

29 **Tomer, M., Kumar, M. (2020).** Improving Text Summarization Using Ensembled Approach Based on Fuzzy with LSTM. Arabian Journal for Science and Engineering, Vol. 45, No. 12, pp. 10743–10754.

30 **Suanmali, L., Salim, N., Binwahlan, M.S. (2009).** Fuzzy Logic Based Method for Improving Text Summarization. arXiv preprint arXiv:0906.4690. doi: 10.48550/arXiv.0906.4690.

31 **Chopade, H.A., Narvekar, M. (2017).** Hybrid Auto Text Summarization Using Deep Neural Network and Fuzzy Logic System. In 2017 International Conference on Inventive Computing and Informatics (ICICI), pp. 52–56. doi: 10.1109/ICICI.2017.8365192.

32 **Goularte, F.B., Nassar, S.M., Fileto, R., Saggion, H. (2019).** A Text Summarization Method Based on Fuzzy Rules and Applicable to Automated Assessment. Expert Systems with Applications, Vol. 115, pp. 264–275. doi: 10.1016/j.eswa.2018.07.047.

33 **Kale, M., Rastogi, A. (2020).** Text-to-Text Pre-Training for Data-to-Text Tasks. Proceedings

of the 13th International Conference on Natural Language Generation, pp. 97–102. doi: 10.18653/v1/2020.inlg-1.14.

34 **Hermann, K.M. et al. (2015).** Teaching Machines to Read and Comprehend. Advances in Neural Information Processing Systems, Vol. 28.

35 **Lin, C.-Y. (2004).** ROUGE: A Package for Automatic Evaluation of Summaries. In Workshop on Text Summarization Branches Out, pp. 74–81.

36 **Nallapati, R., Zhai, F., Zhou, B. (2017).** SummaRuNNer: A Recurrent Neural Network Based Sequence Model for Extractive Summarization of Documents. Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 31, No. 1. doi: 10.1609/aaai.v31i1.10958.

37 **Bahdanau, D., Cho, K., Bengio, Y. (2014).** Neural Machine Translation by Jointly Learning to Align and Translate. arXiv preprint arXiv:1409.0473. doi: 10.48550/arXiv.1409.0473.

38 **Yang, M., Qu, Q., Tu, W., Shen, Y., Zhao, Z., Chen, X. (2019).** Exploring Human-Like Reading Strategy for Abstractive Text Summarization. Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, No. 1, pp. 7362–7369. doi: 10.1609/aaai.v33i01.33017362.

39 **Yao, K., Zhang, L., Du, D., Luo, T., Tao, L., Wu, Y. (2018).** Dual Encoding for Abstractive Text Summarization. IEEE Transactions on Cybernetics. doi: 10.1109/TCYB.2018.2876317.

Articles received on 31/10/2025; accepted on 15/12/2025.
**Corresponding author is Charu Gupta.*