Computer Systems Science & Engineering
DOI: 10.32604/csse.2023.027481
Article

Tech Science Press

Ext-ICAS: A Novel Self-Normalized Extractive Intra Cosine Attention Similarity Summarization

P. Sharmila[1,*], C. Deisy[1] and S. Parthasarathy[2]

[1]Deparment of Information Technology, Thiagarajar College of Engineering, Madurai, India
[2]Deparment of Data Science, Thiagarajar College of Engineering, Madurai, India
*Corresponding Author: P. Sharmila. Email: sharmilapc26@gmail.com

Received: 18 January 2022; Accepted: 08 April 2022

Abstract: With the continuous growth of online news articles, there arises the necessity for an efficient abstractive summarization technique for the problem of information overloading. Abstractive summarization is highly complex and requires a deeper understanding and proper reasoning to come up with its own summary outline. Abstractive summarization task is framed as seq2seq modeling. Existing seq2seq methods perform better on short sequences; however, for long sequences, the performance degrades due to high computation and hence a two-phase self-normalized deep neural document summarization model consisting of improvised extractive cosine normalization and seq2seq abstractive phases has been proposed in this paper. The novelty is to parallelize the sequence computation training by incorporating feed-forward, the self-normalized neural network in the Extractive phase using Intra Cosine Attention Similarity (Ext-ICAS) with sentence dependency position. Also, it does not require any normalization technique explicitly. Our proposed abstractive Bidirectional Long Short Term Memory (Bi-LSTM) encoder sequence model performs better than the Bidirectional Gated Recurrent Unit (Bi-GRU) encoder with minimum training loss and with fast convergence. The proposed model was evaluated on the Cable News Network (CNN)/Daily Mail dataset and an average rouge score of 0.435 was achieved also computational training in the extractive phase was reduced by 59% with an average number of similarity computations.

Keywords: Abstractive summarization; natural language processing; sequence-to-sequence learning (seq2seq); self-normalization; intra (self) attention

1 Introduction

The contemporary web has a massive increase in text content resulting in information overload [1]. Searching for informative data is a time-consuming process. Abstractive summarization is an effective solution for summarizing these data. The primary research on text summarization began in the 1960s [2]. Summarization process can be chiefly categorized as extractive and abstractive [3] methods. An extractive summarization is a method of selecting key words, phrases, and sentences from the source articles.

Abstractive Summarization is the method of shortening the content as a short passage without contriving the entire meaning of the complete document.

Lately, Deep Learning has emerged as a powerful learning technique applied to a variety of Natural Language Processing tasks like Machine Translation [4], Speech Recognition [5], Question-Answering [6], Text Conversation and Summarization [7]. Finaly, abstractive summarization task has been framed as a sequence language modeling task [8] as a standard encoder-decoder framework. In this framework, two Recurrent Neural Networks (RNN) are trained; one of them at the encoder side to encode the input sequences, thereby mapping them into a fixed-size vector as hidden states and the second one is at the decoder meant for decoding the output sequence from the hidden states.

Seq2seq with attention [9,10] was introduced to enhance the summarization task. The purpose of attention is to focus on key sentences with some attention weights assigned to them. Though attention is flexible and highly parallelizable with matrix computation, more memory is required. Hence, a novel method for computing dependency position (DP) has been incorporated into intra-attention. In a given text, ahead of a certain distance, denoted by D, there is no considerable dependency between the sentences. The value of this threshold plays a vital role in the analysis. By doing so, the number of matrix computations gets reduced in self-attention also carried out in parallel. These parallelizable intra-attentions make use of the Graphical Processing Unit (GPU) more efficiently.

So, a novel model has been proposed to overcome these problems by applying self normalized **Ext-ICAS** in extractive methods and then by combining with abstractive sequence methods that make use of parallel computations efficiently. Major contributions of this paper are:

- Development of a novel **Ext-ICAS** in extractive phase to extract relevant sentences with sentence dependency position (DP), which are then built on standard seq2seq encoder-decoder framework.
- Ext-ICAS enables parallel computation with self-normalized intra-cosine attention similarity.
- Dealing with sentence-level representation reduces the computational complexity in the extractive phase since intra-attention works with fixed-size vectors.
- Comparison of the relevant content and abstractness with the existing methods on the CNN/Daily Mail Dataset using Recall-Oriented Understudy for Gisting Evaluation (ROUGE) scores. Also, compared the training loss for the Bi-GRU encoder and Bi-LSTM encoder sequence model.

The rest of this article is organized as follows: Section 2 discusses the related works. The two-phase summarization model is briefly explained in Section 3. Dataset and experimental results are presented in Section 4. Whereas Section 5 gives a detailed description of observations and discussion, conclusions are drawn in Section 6.

2 Related Works

The traditional summarization method is an extractive graph-based ranking algorithm proposed as LEXRANK [11,12]. These algorithms limit syntactic and semantic structure. Therefore, to maintain semantic relation, certain prediction-based neural network models like Word2Vec [13], FastText [14], GloVe were proposed for word embedding. Doc2vec [15] distributed vector representation model was proposed for sentence embedding. Bidirectional Encoder Representations from Transformers (BERT) [16–18] the pre-trained language models in sequence modeling. However, these had limits for extended input sequences and could only train with a maximum of 512 tokens.

Extractive methods are more informative but they lack cohesion [19]. So [20,21] proposed a Convolution Sequence to Sequence (ConvS2S) network with Convolution Neural Network (CNN) as an encoder for large vocabulary but limited in sequence order which requires additional layers. Later, abstractive summarization models were used RNN encoder and decoder for sequence generation tasks. A comparative study between RNN and CNN was presented in [22]; the authors have concluded that RNN is better for long-sequence prediction.

Bi-LSTM [23] was used as an encoder that showed better performance in capturing information than unidirectional gated LSTM [24]. Seq2seq model forced the compression of the input sequences leading to a loss of information. To avoid such losses, [9] proposed the use of different attention such as Dot-Product, Scaled Dot-Product, and Additive at the decoder to extract more informative sentences.

Self or Intra attention [25]was introduced, by focusing within the concerned sentences for the extraction rather than the whole to reduce the number of computations which has the ability of parallel computing and results in better utilization of the GPU.

2.1 Comparison of Seq2Seq Models in Text Summarization

RNN's perform well for sequence data with less expensive and faster training, but has short-term memory problem. So the semantic representation using LSTM and GRU for large documents is relatively complex in respect to more number of parameters and computations, higher memory consumption, and longer training time.GRU required fewer parameters than LSTM and so faster in computations. But LSTM is meant for better input representation, accurate results, and bidirectional network learning features from both past as well as future inputs for better representation and faster convergence.

An overview of different sequence models has been presented in Tab. 1.

Table 1: An overview of seq2seq models for the abstractive summarization

Reference	Highlights	Framework	Dataset	Metrics
[22]	BiLSTM Sequence Modeling	BiLSTM → BiLSTM	CNN/DailyMail	Rouge
[23]	LSTM Sequence Modeling	LSTM → LSTM	CNN/DailyMail	Rouge
[25]	Intra Attention	LSTM → IntraAttention → LSTM	CNN/DailyMail	Rouge
[26]	Topic Aware Attention CNN → CNN	Topic Aware Attention CNN → CNN	CNN/DailyMail, Gigaword	Rouge
[27,28]	BERT Modeling	BiLSTM → Self-Attention → LSTM	CNN/DailyMail	Rouge

2.2 Research Gap

Even though recent advances in seq2seq language modeling techniques on deep learning have achieved a high level of semantic representation and sequence generation, through a meticulous review of the literature, some research gaps were identified.

A deep neural network performs better on a large dataset. The aforementioned seq2seq model with Bi-GRU encoder [29] requires a week to train the model. Hence, high computational complexity is the major challenging task of training sequential inputs. Another research gap evident is the need to preserve all the relevant information in the target summary without redundancies.

Reducing computational complexity and extracting relevant information are the two major bottlenecks that have been addressed in our proposed work. By combining the advantages of extractive and abstractive models in the form of a two-phase deep neural document summarization technique, sentence extraction has been accomplished using a self-normalized parallel process, and summary generation has been carried out through a seq2seq network.

3 Proposed Two-Phase Summarization Model

For summarizing large documents, a two-phase deep neural intra-attention abstractive summarization model is proposed. Fig. 1 shows the workflow of the proposed model. The proposed model comprises two phases, namely extractive and abstractive phases.

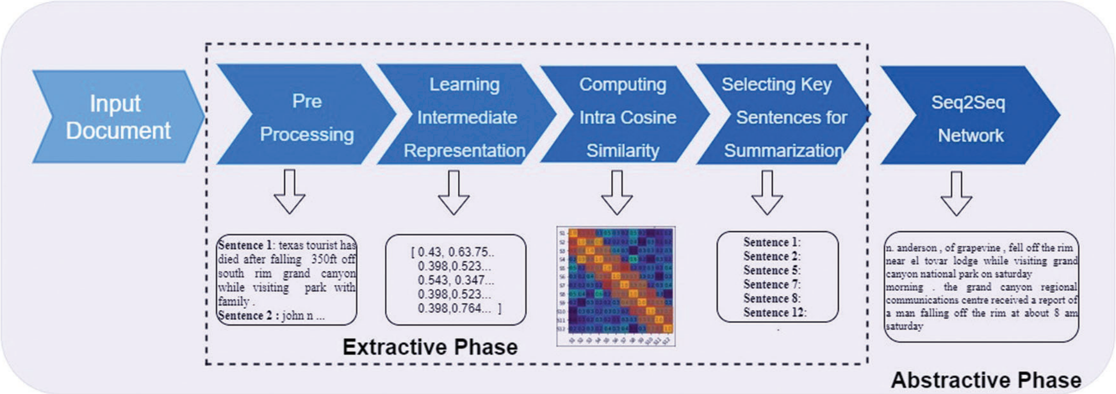

Input Document
Pre Processing
Learning Intermediate Representation
Computing Intra Cosine Similarity
Selecting Key Sentences for Summarization
Seq2Seq Network
Sentence 1: texas tourist has died after falling 350ft off south rim grand canyon while visiting park with family .
Sentence 2 : john n ...
[0.43, 0.63.75..
0.398,0.523...
0.543, 0.347...
0.398,0.523...
0.398,0.764...]
Sentence 1:
Sentence 2:
Sentence 5:
Sentence 7:
Sentence 8:
Sentence 12:
n. anderson , of grapevine , fell off the rim near el tovar lodge while visiting grand canyon national park on saturday morning . the grand canyon regional communications centre received a report of a man falling off the rim at about 8 am saturday
Extractive Phase
Abstractive Phase

Figure 1: Workflow of the proposed two-phase deep neural summarization model

3.1 Extractive Phase

The objective of the proposed extractive phase is to extract more informative and relevant sentences to build an extractive summary using the self-normalized Ext-ICAS method. Self-normalized cosine attention with dependency position is the novelty proposed in the extractive phase for parallel computation. Intra-attention is a feed-forward neural network that enables parallel processing and hence makes use of the GPU efficiently. Compared with RNN and gated RNN, it converges quickly with a minimum number of parameters. Fig. 2 explains the steps and layers of the extractive phase.

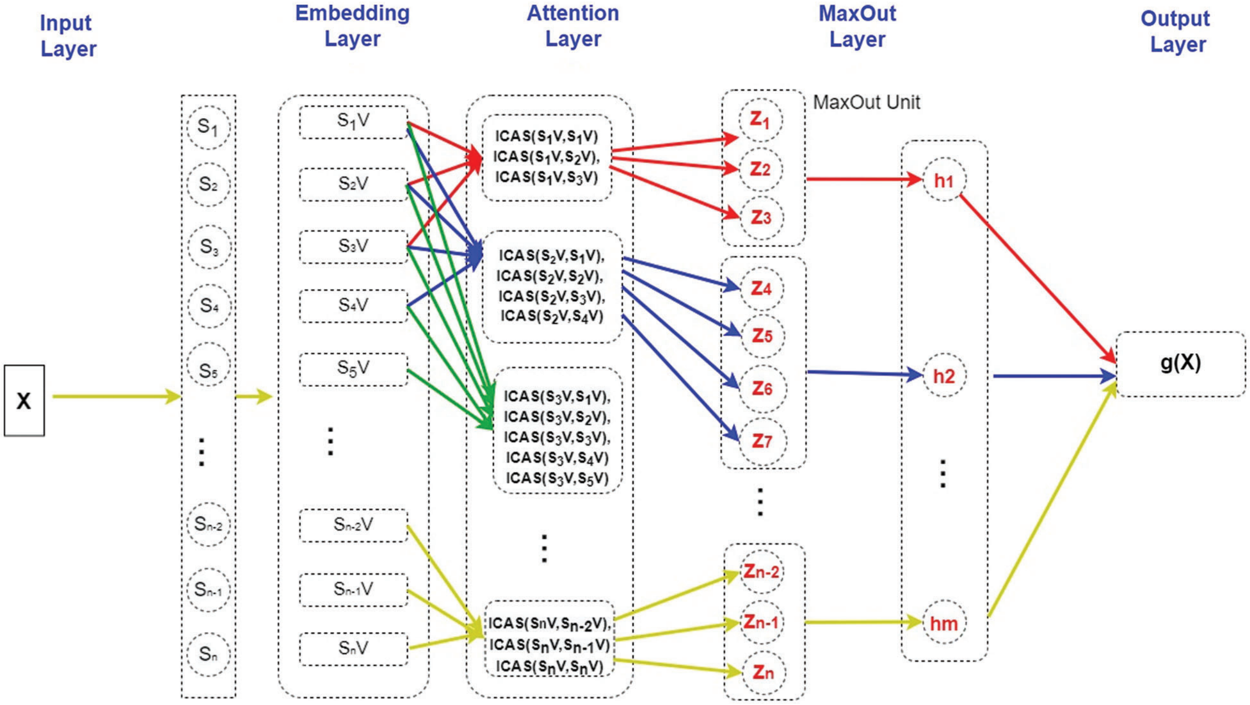


Figure 2: Layers of the extractive phase with self-normalized intra-attentions

3.1.1 Self-Normalized Cosine Attention

Computing cosine itself self-normalized one, the range of output ranges from 0 to 1, hence there is no need for explicit normalization and activation functions as shown in Fig. 3. S_iV and $_{Si+1}$V are two sentence vectors from the input X document. Intra Cosine Self Normalization is computed.

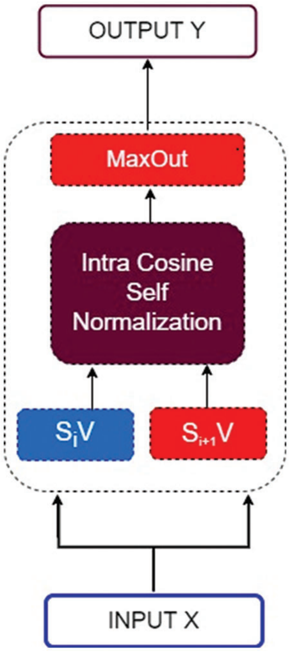


Figure 3: Individual self-normalized similarity computation model

The extractive phase has five steps. They are explained here:

a) **Input layer (Preprocessing):** The input documents are split into sentences. The input document X = {S_1, S2, S3…S_n}.

b) **Embedding layer (Learning Intermediate Representation):** The sentences are embedded into sentence vectors using doc2vec, which is effective in capturing semantic structure in the learning vector representation. X = {S_1V, S2V, S3V …S_nV}

$$X_i = S_i\text{V}; \tag{1}$$

c) **Attention layer (Computing Self Normalized Intra-Cosine Attention Similarity):** Computing self- normalized Intra Cosine Attention similarity score between each sentence vector. Intra-attention at the word level requires more matrix computations, so in this model, the sentence level attention has been employed to distill key sentences and reduce the data size, thereby resulting in minimum computations.

$$ICAS = \left\{ \begin{array}{cc} f\left(S_i, S_j\right) & i \neq j \\ 0 & otherwise \end{array} \right. \tag{2}$$

$$f\left(S_i, S_j\right) = \sum_{i=1}^{n} \sum_{j=1}^{DP} \cos\left(s_i, s_j\right) \tag{3}$$

where, S_i and S_j are fixed-size vector embeddings obtained using doc2vec; n is the number of sentences in the document; DP is the sentence dependency position; When i = j, it means the same sentence in the document, and hence similarity will be at the highest level, that is 1, leading to duplication. Therefore such cases need to be eliminated.

d) Similarities between the sentences are computed using the ICAS method with sentence dependency position DP.

$$Z_{ij} = ICAS\left(S_iV, S_jV\right) \tag{4}$$

e) **Max-Out Layer** $\boldsymbol{Z}(\boldsymbol{Xi})$**: S**imply, the max of inputs. This max-out is modified by having only the advantages of a Rectified Linear Unit (ReLU) unit and just acting as a dropout layer.

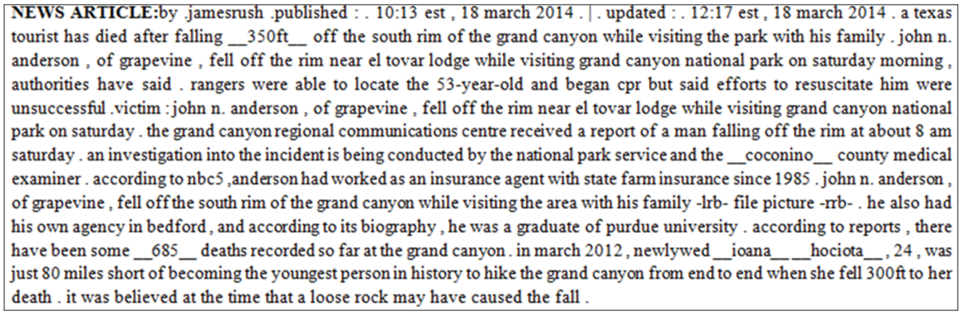
$$h_i(X) = Max\left(Z_{ij}\right) \tag{5}$$

f) **Sentence Selection or Output layer g(X):** Sentences with high similarity are extracted in the extractive phase. g(X) is the concatenation of the maximum attention score.

$$g(X) = \sum h_i(X) \tag{6}$$

3.1.2 Sample Output

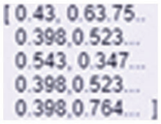
a) First, the input news article is split into 12 sentences. S1 to S12 shown in Fig. 4



NEWS ARTICLE:by .jamesrush .published : . 10:13 est , 18 march 2014 . | . updated : . 12:17 est , 18 march 2014 . a texas tourist has died after falling __350ft__ off the south rim of the grand canyon while visiting the park with his family . john n. anderson , of grapevine , fell off the rim near el tovar lodge while visiting grand canyon national park on saturday morning , authorities have said . rangers were able to locate the 53-year-old and began cpr but said efforts to resuscitate him were unsuccessful .victim : john n. anderson , of grapevine , fell off the rim near el tovar lodge while visiting grand canyon national park on saturday . the grand canyon regional communications centre received a report of a man falling off the rim at about 8 am saturday . an investigation into the incident is being conducted by the national park service and the __coconino__ county medical examiner . according to nbc5 ,anderson had worked as an insurance agent with state farm insurance since 1985 . john n. anderson , of grapevine , fell off the south rim of the grand canyon while visiting the area with his family -lrb- file picture -rrb- . he also had his own agency in bedford , and according to its biography , he was a graduate of purdue university . according to reports , there have been some __685__ deaths recorded so far at the grand canyon . in march 2012 , newlywed __ioana____hociota__ , 24 , was just 80 miles short of becoming the youngest person in history to hike the grand canyon from end to end when she fell 300ft to her death . it was believed at the time that a loose rock may have caused the fall .

Figure 4: Sample news article

b) Learning Intermediate Representation: Sentences are converted into a vector using doc2vec gensim shown in Fig. 5



```
[ 0.43, 0.63.75..
  0.398,0.523...
  0.543, 0.347...
  0.398,0.523...
  0.398,0.764... ]
```

Figure 5: Sentence vector

c) Computing self normalized Intra-Cosine Attention Similarity with Dependency Position DP as 2 Fig. 6. Self-normalized attention scores for Sentence S1 with S0, S1, S2, and S3. So, attention scores ICAS for S1 with the previous two sentences, and the next sentences are calculated.

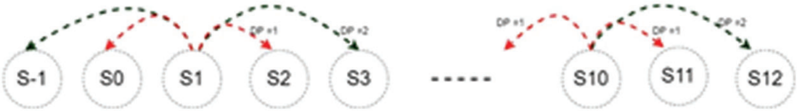


Figure 6: Sentence similarities with DP as 2

d) Self-Normalized Attention Score: Tab. 2 shows the Self Normalized Attention Score ICAS with all sentences in a document. Similarity among the same sentence in the document will be 1, leading to duplication. Therefore, such comparison needs to be eliminated; to minimize the computations which are highlighted in Tab. 2

Table 2: Self normalized attention score

Self normalized attention score	Sentence dependency DP = 2	Max-Out h(x)	Selected sentences	g(X)	Selected sentences	Max-Out h(x)	Sentence dependency DP = 2	Self normalized attention score

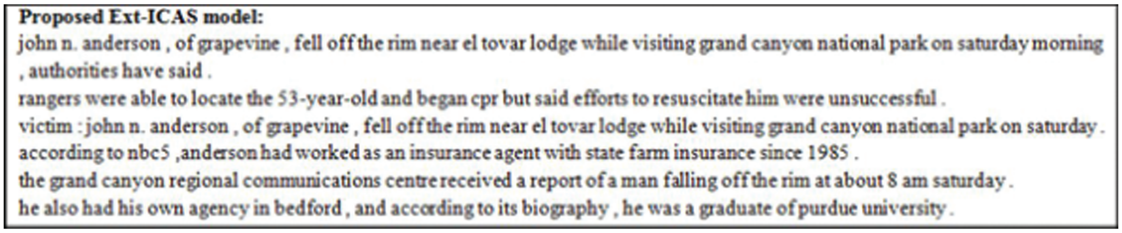
ICAS(S_1V,S_1V)
ICAS(S_1V,S_2V),
ICAS(S_1V,S_3V)
	1
	0.2
	0.1
		0.2	S1, S2	S2 S3 S4 S5 S7 S9	S7, S9	0.4	00
							0.3
							1
							0.3
							0.4
								ICAS(S_7V,S_5V),
								ICAS(S_7V, S_6V),
								ICAS(S_7V, S_7V),
								ICAS(S_7V,S_8V),
								ICAS(S_7V,S_9V),

ICAS(S_2V,S_1V), ICAS(S_2V, S_2V), ICAS(S_2V,S_3V), ICAS(S_2V,S_4V)	0.1 1 0.4 0.9	0.9	S2, S4		S6, S7, S8, S9	0.3	0.3 0.3 1 0.3 0.1	ICAS(S_8V, S_6V), ICAS(S_8V, S_7V), ICAS(S_8V, S_8V), ICAS(S_8V,S_9V), ICAS(S_8V,S_{10}V),
ICAS(S_3V,S_1V), ICAS(S_3V,S_2V), ICAS(S_3V,S_3V), ICAS(S_3V,S_4V) ICAS(S_3V,S_5V)	0.1 0.4 1 0.3 0.4	0.4	S2, S3, S5		S7, S9	0.4	0.4 0.3 1 0.2 0.3	ICAS(S_9V, S_7V), ICAS(S_9V, S_8V), ICAS(S_9V, S_9V), ICAS(S_9V,S_{10}V), ICAS(S_9V,S_{11}V),
ICAS(S_4V,S_1V), ICAS(S_3V,S_2V), ICAS(S_4V,S_3V), ICAS(S_4V,S_4V) ICAS(S_4V,S_5V)	0.9 0.3 1 0.3 .1	0.9	S1,S4		S11, S12	0.3	0.1 0.2 1 0.3 0.3	ICAS(S_{10}V, S_8V), ICAS(S_{10}V, S_9V), ICAS(S_{10}V,S_{10}V), ICAS(S_{10}V,S_{11}V), ICAS(S_{10}V,S_{12}V),
ICAS(S_5V, S_3V), ICAS(S_5V, S_4V), ICAS(S_5V, S_5V), ICAS(S_5V,S_6V) ICAS(S_5V,S_7V)	0.2 0.3 1 0.2 0.0	0.3	S4, S5		S9, S10, S11	0.3	0.3 0.3 1 0.2	ICAS(S_{11}V, S_9V), ICAS(S_{11}V,S_{10}V), ICAS(S_{11}V,S_{11}V), ICAS(S_{11}V,S_{12}V),

e) Max-out function h(x) is used to extract the sentences with maximum scores, and then function g(x) concatenates all the selected sentences as extractive summaries. No threshold value needs to be given manually to this function. Also [29] explain max-out function works better than ReLU. Sentences S2, S3, S4, S5, S7, and S9 are extracted from the news article as an extractive summary shown in Fig. 7, and the comparison of our proposed Ext-ICAS extractive model with Sentence Dependency Attention is shown in Fig. 8.

3.2 Abstractive Phase

Sentences from the extractive phase are fed into the seq2seq framework abstractive phase to generate a quality summary. Fig. 9 shows the illustration of the abstractive phase encoder-decoder model with attention. Extracted sentences from the Ext-ICAS model fed into the seq2seq abstractive phase. Sentences are embedded into a vector using a Bi-LSTM encoder and attention weights are computed using alignment and context vector. LSTM decoder generates the summary sequence.



Proposed Ext-ICAS model:

john n. anderson , of grapevine , fell off the rim near el tovar lodge while visiting grand canyon national park on saturday morning , authorities have said .

rangers were able to locate the 53-year-old and began cpr but said efforts to resuscitate him were unsuccessful .

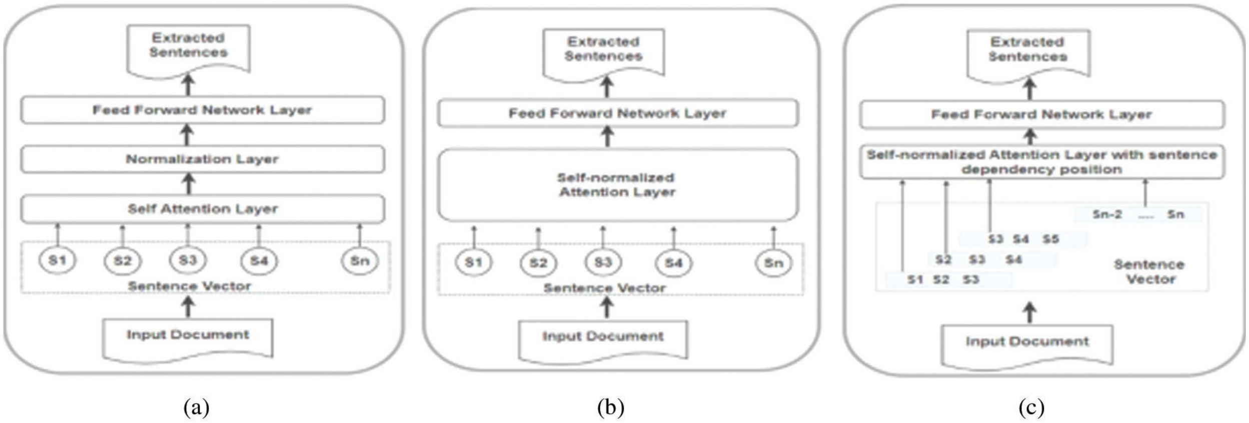
victim : john n. anderson , of grapevine , fell off the rim near el tovar lodge while visiting grand canyon national park on saturday .

according to nbc5 ,anderson had worked as an insurance agent with state farm insurance since 1985 .

the grand canyon regional communications centre received a report of a man falling off the rim at about 8 am saturday .

he also had his own agency in bedford , and according to its biography , he was a graduate of purdue university .

Figure 7: Extracted sentences



(a) (b) (c)

Figure 8: a) Self-attention with explicit normalization layer b) proposed Ext-ICAS self-normalized model c) proposed Ext-ICAS self-normalized model with sentence dependency position model

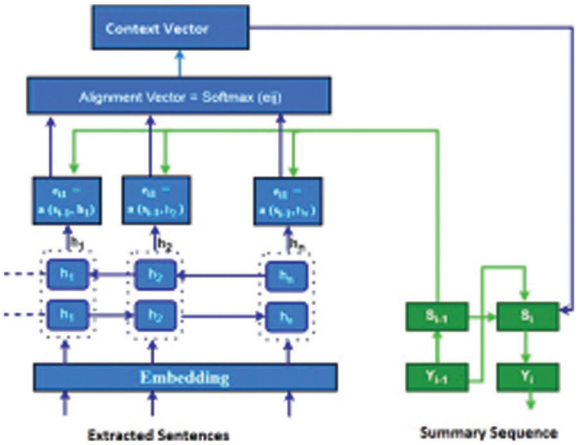

Context Vector
Alignment Vector = Softmax (eij)
$e_{i1} = a(s_{i-1}, h_1)$
$e_{i2} = a(s_{i-1}, h_2)$
$e_{in} = a(s_{i-1}, h_n)$
h_1
h_2
h_n
S_{i-1}
S_i
Y_{i-1}
Y_i
Embedding
Extracted Sentences
Summary Sequence

Figure 9: Abstractive phase encoder-decoder model with attention

Following are the steps in the abstractive phase:

1. **Encoder**: With Bi-LSTM as the encoder, all sentences are encoded into a hidden vector representation.
2. **Attention**: Focuses on all the hidden vectors to select the keywords.
3. **Decoder**: Using LSTM beam-search, the decoder generates a summary sequence from the extracted words.

3.2.1 Encoder

Bi-LSTM supersedes LSTM in terms of reduced convergence cycle (i.e., in both forward and backward direction). Thus Bi-LSTM has been chosen as the encoder. Eqs. (7)–(9) provide formulae for forward, backward, and concatenated-LSTM respectively.

$$\overrightarrow{h_t} = \overrightarrow{biLstm}\left(v_t, \overrightarrow{h_{t-1}}\right) \tag{7}$$

$$\overleftarrow{h_t} = \overleftarrow{biLstm}\left(v_t, \overleftarrow{h_{t-1}}\right) \tag{8}$$

$$h_t = \left[\overrightarrow{h_t}, \overleftarrow{h_t}\right] \tag{9}$$

where, v_t is the input vector, h_{t-1} and h_t are the hidden vector for the previous state and current state;

$\overrightarrow{h_t}$ and $\overleftarrow{h_t}$ are the forward and hidden vector; h_t is the concatenation of the hidden state vector of both $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$.

3.2.2 Attention

It focuses on relevant words for the whole input sequence by assigning each word with a weighted score. These weights are then aggregated with hidden vectors of both the encoder and decoder. As a result, the key concepts are represented in the form of a feature vector. This can be computed using Eq. (10).

$$e_i = \tanh\left\{h_i^{enc}, h_i^{dec}\right\} \tag{10}$$

where, e_i is the feature of the ith word; h_i^{enc} , h_i^{dec} is the hidden vector from the encoder and decoder of the ith word respectively;

$$\alpha_i = softmax(e_i) = \frac{\exp(e_i)}{\sum_k \exp(e_n)} \tag{11}$$

where αi is the alignment vector from the encoder. Then the alignment vector for each word can be arrived at using Eq. (11). From the alignment vector obtained in the previous step, these words, which meet certain threshold values are selected in the form of a context vector using Eq. (12).

$$context\ vector = \sum \alpha_i h_i^{enc} \tag{12}$$

3.2.3 Decoder

In the decoder, the output summary is predicted as sequences of words using a searching algorithm. The extracted feature vector from the attention layer is fed as input for the decoder to generate a summary, using the Naive Bayes method by maximizing conditional probability with beam size 2. As beam size increases beyond 2, the results will be more optimized but with more memory consumption. Since the beam size of 2 results in reduced search space, it has been chosen.

As a post-processing step, the proposed model incorporates the pointer mechanism [30] and copy mechanism [31], to avoid Out-of-Vocabulary (OoV) word problem.

4 Experiment and Results

4.1 Dataset

To evaluate the proposed model, the benchmark dataset CNN/Daily Mail is adopted by most of the existing methods. Initially, the dataset is framed for question-answering [32]. Later, it is extended for the summarization task as well. This dataset contains over 300 K unique news articles, each article with an average of 800 words, with a highlighted summary of 3 to 4 lines each. The training/validation/testing split was 287,113/13/368/11490. Link: https://www.kaggle.com/gowrishankarp/newspaper-text-summarization-cnn-dailymail.

4.2 Experimental Setup

In the extractive phase, preprocessed sentence embeddings are implemented using gensim doc2vec, a single-layer feed-forward neural network as an intra-attention layer is built. Then, the intra-cosine similarity score is computed with DP as 2. No explicit normalization is implemented since the model itself is self-normalized.

The abstractive model has 256-dimensional hidden states and 128-dimensional word embeddings. A deep neural network works with fixed size vocabulary to generate text sequences. So vocab size is fixed as 50 K (the top frequent words) in this model. This will enable the framework to preserve relevant information. As could be seen from Fig. 10, for a vocal size up to 50 k words, the training loss is at a minimal level; beyond that a spurt in training loss. Hence, the vocal size has been fixed as 50 K words.

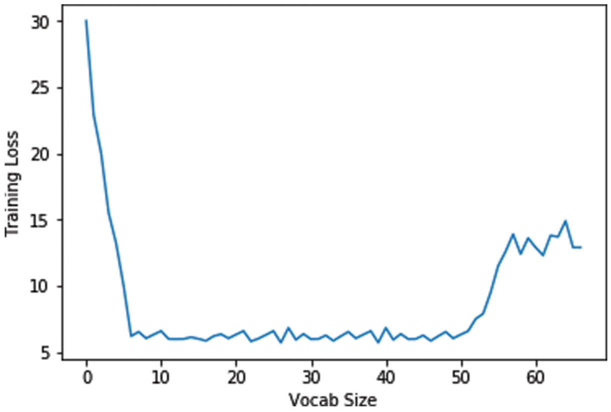


Figure 10: Training loss *vs*. Vocab size

Seq2seq framework has been modeled as a two-layered Bi-LSTM encoder with attention [4] and LSTM decoder with beam size as 2. The learning rate has been maintained as 0.15.

4.3 Evaluation Metrics

The quality of a summary can be evaluated by human beings depending on its saliency, fluency, redundancy, and grammar. This evaluation varies from one human annotator to the other. So, it is decided to evaluate the using Rouge [33] scores.

Rouge is a package introduced in the framework for measuring the quality of summary by counting the number of word overlapping or sentence similarities. Rouge-n for n-gram similarity and Rouge - L for longest sequences are computed using Eq. (13);

$$rouge = \frac{\sum_{S\epsilon\{ref\}} \sum_{n=grams} count_{match}(n - gram)}{\sum_{S\epsilon\{ref\}} \sum_{n=grams} count\ (n - gram)} \tag{13}$$

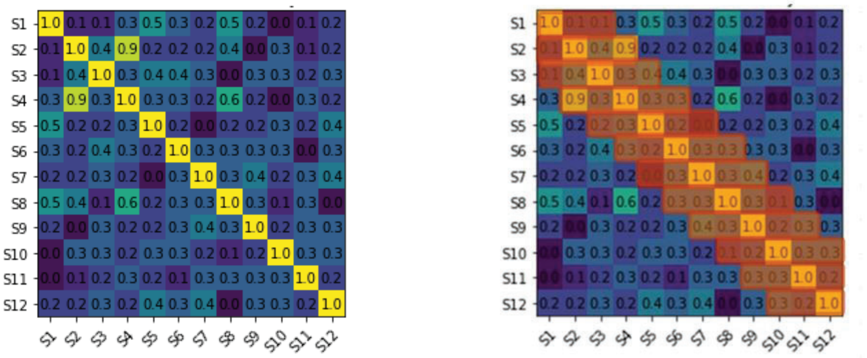
where ref is the reference summary; count $_{match}$ (n-gram) is the number of n-gram matches between reference summary and model generated summary; count (n-gram) is the number of the n-gram in reference summary.

5 Results and Discussion

This section presents a comparative study of the proposed **Ext-ICAS** summarization model with the existing extractive and abstractive summarization models. At the outset, it is observed that **Ext-ICAS** extractive phase summary extracts semantically related sentences with a minimum number of similarity computations. Compared with RNN, feed-forward layers have a less number of parameters and also support parallel computation, thus the time complexity is reduced.

For example, let us consider a news article with N sentences; its similarity computation without DP is **N*N**, whereas, when DP is introduced, the same gets reduced to **N*(DP*2)**, where N is the number of sentences in the document, DP is the sentence dependency position and the numeric value of 2 signifies the fact that the previous and next sentences are taken into consideration.

Let a randomly-selected news article with 12 sentences from the chosen dataset be taken for visualization purposes; the similarity score between the sentence vectors is computed as shown in Fig. 11. Fig. 11 a) shows the similarity score using Ext-ICAS model without dependency position and Fig. 11 b) shows the similarity score using Ext-ICAS model with sentence dependency position (DP = 2). The diagonal elements always take a value of 1, which means that they represent either duplicates or highly related sentences and so the diagonal values are not considered for further computation. After a certain distance [25], the dependencies between the sentences are meaningless. So, the parameter DP is introduced to reduce the number of computations.



(a) Ext-ICAS without dependency position (b) Ext-ICAS with dependency position as 2

Figure 11: (a) Ext-ICAS without dependency position (b) Ext-ICAS with dependency position as 2

By tuning the values of DP, rouge scores are calculated. From Tab. 3, it is observed that the R1 score remains constant after DP ≥ 3 and small variation between DP values as 2 and 3. So, in the proposed model, we fix the DP as 2. The number of sentences N is 12. Similarity, computation without DP is 144, (i.e., 12 × 12); with DP = 2 it is 48, (i.e., 12 × 2 × 2). Thus two-thirds of similarity computations are evaded, which reduces the computations by 60%.

Table 3: Effects of sentence dependency position with ROUGE scores on CNN/Daily mail dataset

Sentence dependency position DP	Number of sentences selected (%)	Ext-ICAS Extractive phase rouge score on CNN/ DailyMail dataset		
		R-1	R-2	R-3
Without DP	100	0.401	0.142	0.309
DP = 2	41.66	0.411	0.158	0.346
DP = 3	66.66	0.412	0.155	0.321
DP = 4	66.66	0.412	0.155	0.321
DP > 4	75	0.412	0.151	0.330

5.1 Impact on Enriched Vocabulary

Extracted relevant informative sentences from the proposed Ext-ICAS model enriched the vocabulary with the most informative and relevant keywords as fixed-size vocab (top 50K words) without any repetition. Summary generated from extracted sentences avoids redundancies. Hence, Ext-ICAS combined with the seq2seq abstractive phase has enhanced the performance.

5.2 Comparison with Existing Model

Time consumption is the biggest challenge with continuously growing data. Now the minimized data had shown improved performance in training. From Fig. 12, it could be observed that the proposed model Ext-ICAS + PGN (Bi-GRU encoder)converged at around 140K iterations and Ext-ICAS + ABS (with Bi-LSTM encoder) converged around 160K iterations with minimum loss intraining. But the baseline PGN model with Bi-GRU encoder converges after 200K iterations with lots of loss fluctuations, which leads to slowing-down of the training process.

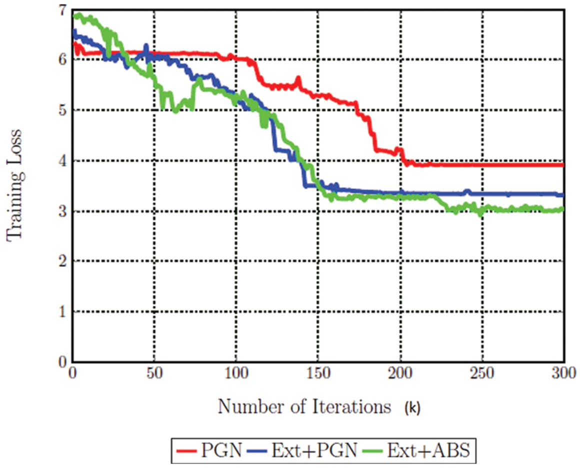


Figure 12: Training loss *vs*. Number of iteration variations

The **Ext-ICAS** extractive phase summary extracts semantically related sentences and achieves better Rouge scores than the existing models. In combination with the abstractive phase also it has performed better in the Rouge scores. In Tab. 4, Rouge scores R-1, R-2, and R-L are compared with the existing extractive summarization models: LEAD-N, LEXRANK and (Hierarchical Structured Self-Attentive Model for Extractive Document Summarization) HSSAS and Pointer-Generator Networks (PGN). Tab. 5 illustrates the comparative study of the Rouge scores of R-1, R-2, and R-L with the existing abstractive summarization models (ABS).

Table 4: Comparison of rouge scores on CNN/Daily mail dataset for extractive summarization models

Extractive summarization models	Summarization techniques	ROUGE		
		R-1	R-2	R-L
Lead-N	Top N sentences as summary	0.392	0.157	0.355
Lexrank	Graph-based ranking Algorithm	0.398	0.15	0.344
HSSAS	Hierarchical self-attention at both word and sentence level	0.412	0.174	0.364
Ext-ICAS (Proposed model)	Ext-ICAS with sentence DP	**0.431**	**0.182**	**0.3661**

Table 5: Comparison of rouge scores on CNN/Daily mail dataset for abstractive summarization models

Abstractive summarization models	Summarization techniques	ROUGE		
		R-1	R-2	R-L
ABS, ABS+	ConvS2S - Convolution Encoder and RNN Decoder	0.312	0.253	0.311
PGN	Bi-GRU Encoder incorporates copy and coverage mechanism	0.395	0.133	0.315
Ext-ICAS + PGN	Ext-ICAS with sentence DP + Bi-GRU Encoder	0.435	0.145	0.359
Ext-ICAS + Abstractive (Proposed two-phase model)	Ext-ICAS with sentence DP + Bi-LSTM Encoder	**0.441**	**0.176**	0.348

The proposed model yields a high R-1 score, thereby maintaining most of the relevant informative keywords than the other baseline models. R-2 also attained more or less comparable results by maintaining co-occurrences of bigrams. The extractive model achieved a high R-L score by keeping the extracted sentence intact. But, in the abstractive phase, the R-L score is less in comparison with the extractive phase because of the rephrasing of the sentences in the abstractive summarization phase.

Graphical representation of rouge scores for both models is shown in Fig. 13. Comparatively high R1, R2 rouge score at extractive phase than abstractive combine with Ext-ICAS summarization model.

The proposed model yields a high R-1 score, thereby maintaining most of the relevant informative keywords than the other baseline models. R-2 also attained more or less comparable results by maintaining co-occurrences of bigrams. The extractive model achieved a high R-L score by keeping the extracted sentence intact. But, in the abstractive phase, the R-L score is less in comparison with the extractive phase because of the rephrasing of the sentences in the abstractive summarization phase.

5.3 Generated Summaries

Sample summaries generated by the existing extractive and abstractive model compared with the proposed Ext-ICAS extractive model and also combined with the abstractive model are shown in Fig. 14 with the reference summary.

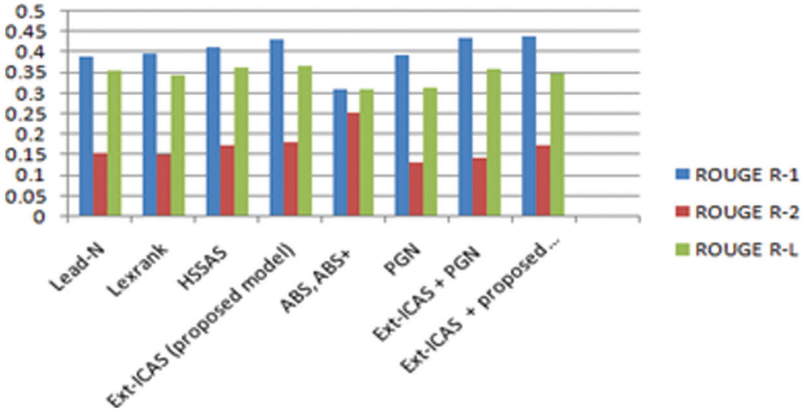
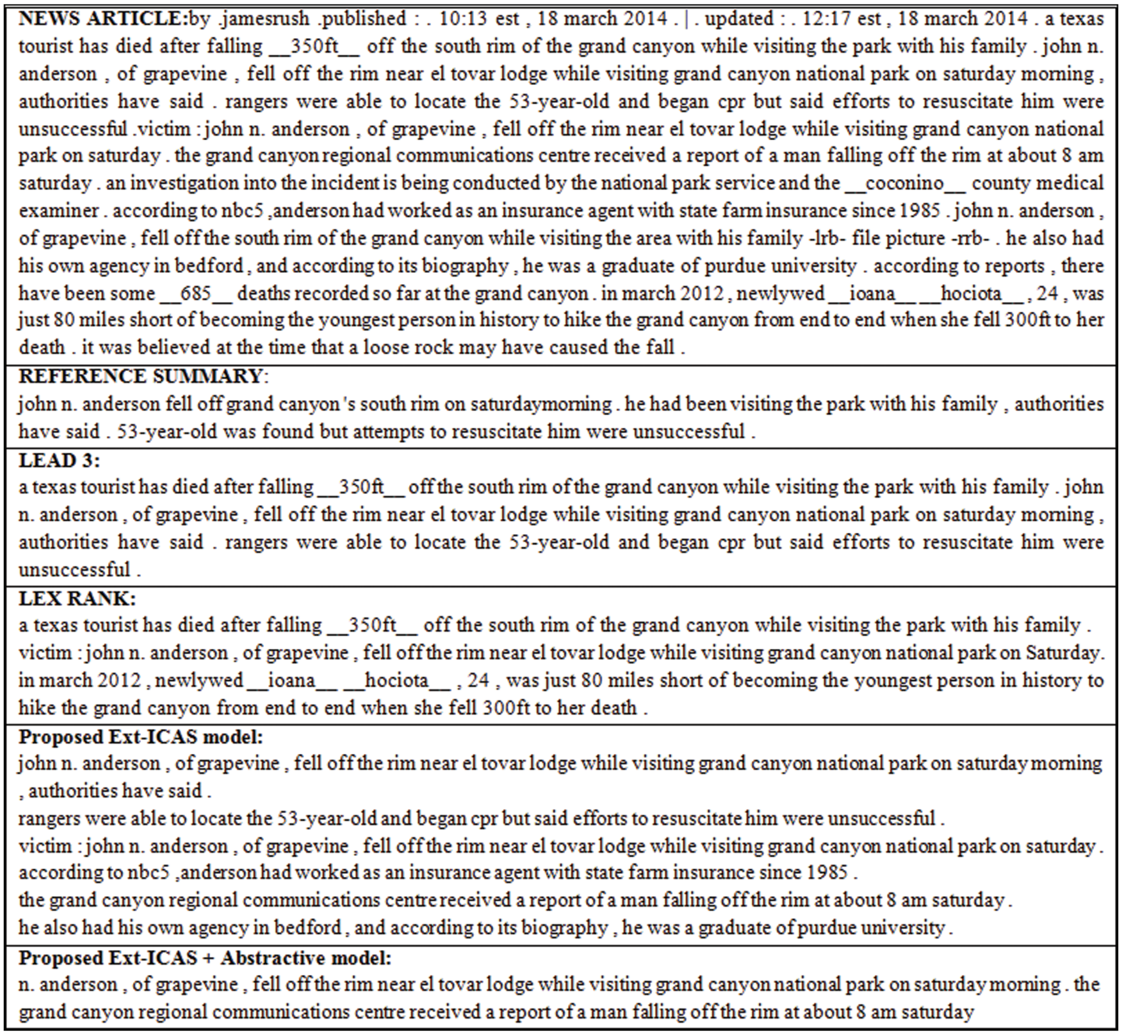


Figure 13: Comparative rouge scores (R-1, R-2, R-L) on CNN/Daily mail dataset



NEWS ARTICLE:by .jamesrush .published : . 10:13 est , 18 march 2014 . | . updated : . 12:17 est , 18 march 2014 . a texas tourist has died after falling __350ft__ off the south rim of the grand canyon while visiting the park with his family . john n. anderson , of grapevine , fell off the rim near el tovar lodge while visiting grand canyon national park on saturday morning , authorities have said . rangers were able to locate the 53-year-old and began cpr but said efforts to resuscitate him were unsuccessful .victim : john n. anderson , of grapevine , fell off the rim near el tovar lodge while visiting grand canyon national park on saturday . the grand canyon regional communications centre received a report of a man falling off the rim at about 8 am saturday . an investigation into the incident is being conducted by the national park service and the __coconino__ county medical examiner . according to nbc5 ,anderson had worked as an insurance agent with state farm insurance since 1985 . john n. anderson , of grapevine , fell off the south rim of the grand canyon while visiting the area with his family -lrb- file picture -rrb- . he also had his own agency in bedford , and according to its biography , he was a graduate of purdue university . according to reports , there have been some __685__ deaths recorded so far at the grand canyon . in march 2012 , newlywed __ioana__ __hociota__ , 24 , was just 80 miles short of becoming the youngest person in history to hike the grand canyon from end to end when she fell 300ft to her death . it was believed at the time that a loose rock may have caused the fall .

REFERENCE SUMMARY:
john n. anderson fell off grand canyon 's south rim on saturdaymorning . he had been visiting the park with his family , authorities have said . 53-year-old was found but attempts to resuscitate him were unsuccessful .

LEAD 3:
a texas tourist has died after falling __350ft__ off the south rim of the grand canyon while visiting the park with his family . john n. anderson , of grapevine , fell off the rim near el tovar lodge while visiting grand canyon national park on saturday morning , authorities have said . rangers were able to locate the 53-year-old and began cpr but said efforts to resuscitate him were unsuccessful .

LEX RANK:
a texas tourist has died after falling __350ft__ off the south rim of the grand canyon while visiting the park with his family .
victim : john n. anderson , of grapevine , fell off the rim near el tovar lodge while visiting grand canyon national park on Saturday.
in march 2012 , newlywed __ioana__ __hociota__ , 24 , was just 80 miles short of becoming the youngest person in history to hike the grand canyon from end to end when she fell 300ft to her death .

Proposed Ext-ICAS model:
john n. anderson , of grapevine , fell off the rim near el tovar lodge while visiting grand canyon national park on saturday morning , authorities have said .
rangers were able to locate the 53-year-old and began cpr but said efforts to resuscitate him were unsuccessful .
victim : john n. anderson , of grapevine , fell off the rim near el tovar lodge while visiting grand canyon national park on saturday .
according to nbc5 ,anderson had worked as an insurance agent with state farm insurance since 1985 .
the grand canyon regional communications centre received a report of a man falling off the rim at about 8 am saturday .
he also had his own agency in bedford , and according to its biography , he was a graduate of purdue university .

Proposed Ext-ICAS + Abstractive model:
n. anderson , of grapevine , fell off the rim near el tovar lodge while visiting grand canyon national park on saturday morning . the grand canyon regional communications centre received a report of a man falling off the rim at about 8 am saturday

Figure 14: Comparison of sample summaries generated by Ext-ICAS model, Ext-ICAS +Abstractive model with LEAD 3, model and LEXRANK model

6 Conclusion and Suggestions for Future Work

In this work, the Ext-ICAS self-normalized deep neural document summarization model for relevant informative summarization has been proposed. The effectiveness of the proposed model both in extractive and abstractive summarization has been evaluated on the CNN/DailyMail dataset. The proposed Ext-ICAS self-normalized deep neural document summarization model incorporated with sentence dependency position performed better in capturing semantic relationships within the documents. The experimental results have shown significant improvement in performance with minimum training time and abstractive relevant sequence extraction. Concerning readability and relevance, it is found to give complete meaning.

Upon evaluation of the proposed model, it was observed that it outperforms the baseline model in terms of ROUGE scores and also in computational training with an average number of similarity computations in the extractive phase being reduced by 59%; the model achieved ROUGE scores of 0.435, 0.176 and 0.348 respectively for R-1, R-2, and R-L.

There is always a tradeoff between beam size and memory consumption, which can be dealt with in future studies to generate abstractive summary. Further, this work can be extended for multi-document summarization and query-based summarization. Later on, the proposed Ext-ICAS model could also be applicable in the abstractive phase. It could also be observed that, in the absence of a reference summary, the rouge score may not be adequate to evaluate the summary quality. In such a situation, further research is needed to look for alternative evaluation metrics.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] M. Yousefi-Azar and L. Hamey, “Text summarization using unsupervised deep learning,” *Expert System with Applications*, vol. 68, no. 1, pp. 93–105, 2017.

[2] M. F. Mridha, A. A. Lima, K. Nur, S. C. Das, M. Hasan *et al.,* “A survey of automatic text summarization: Progress, process and challenges,” *IEEE Access*, vol. 9, pp. 156043–156070, 2021.

[3] J. Pilault, R. Li, S. Subramanian and C. Pal, “On extractive and abstractive neural document summarization with transformer language models,” in *Proc. Conf. on Empirical Methods in Natural Language Processing*, pp. 9308–9319, 2020.

[4] J. Su, J. Zeng, D. Xiong, Y. Liu, M. Wang *et al.,* “A hierarchy-to-sequence attentional neural machine translation model,” *IEEE/ACM Transactions Audio, Speech, Language Processing*, vol. 26, no. 3, pp. 623–632, 2018.

[5] Z. Song, “English speech recognition based on deep learning with multiple features,” *Computing*, vol. 102, no. 3, pp. 663–682, 2020.

[6] L. Q. Cai, M. Wei, S. T. Zhou and X. Yan, “Intelligent question answering in restricted domains using deep learning and question pair matching,” *IEEE Access*, vol. 8, pp. 32922–32934, 2020.

[7] J. Torres, C. Vaca, L. Terán and C. L. Abad, “Seq2seq models for recommending short text conversations,” *Expert System with Applications*, vol. 150, no. 8, pp. 113270, 2020.

[8] Z. Liang, J. Du and C. Li, “Abstractive social media text summarization using selective reinforced Seq2Seq attention model,” *Neurocomputing*, vol. 410, pp. 432–440, 2020.

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones *et al.,* “Attention is all you need,” *Proceedings of Advances in Neural Information Processing Systems*, vol. 2, pp. 5999–6009, 2017.

[10] C. Chootong, T. K. Shih, A. Ochirbat, W. Sommool and Y. Y. Zhuang, “An attention enhanced sentence feature network for subtitle extraction and summarization,” *Expert System with Application*, vol. 178, pp. 114946, 2021.

[11] G. Erkan and D. R. Radev, “LexRank: Graph-based lexical centrality as salience in text summarization,” *Journal of Artificial Intelligence Research*, vol. 22, pp. 457–479, 2004.

[12] W. S. El-Kassas, C. R. Salama, A. A. Rafea and H. K. Mohamed, “EdgeSumm: Graph-based framework for automatic text summarization,” *Information Processing and Management*, vol. 57, no. 6, pp. 102264, 2020.

[13] T. Mikolov, K. Chen, G. Corrado and J. Dean, “Efficient estimation of word representations in vector space, arXiv,” arXiv: 1301.3781, 2013.

[14] Z. Lin, L. Wang, X. Cui and Y. Gu, “Fast sentiment analysis algorithm based on double model fusion,” *Computer Systems Science and Engineering*, vol. 36, no. 1, pp. 175–188, 2021.

[15] Q. Le and T. Mikolov, “Distributed representations of sentences and documents,” in *Proc. of Int. Conf. on Machine Learning*, pp. 1188–1196, 2014.

[16] M. Moradi, G. Dorffner and M. Samwald, “Deep contextualized embeddings for quantifying the informative content in biomedical text summarization,” *Computer Methods and Programs in Biomedicine*, vol. 184, pp. 105117, 2020.

[17] J. Devlin, M. W. Chang, K. Lee and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, pp. 4171–4186, 2019.

[18] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim *et al.,* “BioBERT: A pre-trained biomedical language representation model for biomedical text mining,” *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, 2020.

[19] K. Nandhini and S. R. Balasundaram, “Improving readability through extractive summarization for learners with reading difficulties,” *Egyptian Informatics Journal*, vol. 14, no. 3, pp. 195–204, 2013.

[20] J. P. A. Vieira and R. S. Moura, “An analysis of convolutional neural networks for sentence classification,” in *Proc. of XLIII Latin American Computer Conf.*, pp. 1–5, 2017.

[21] H. Kameoka, K. Tanaka, D. Kwaśny, T. Kaneko and N. Hojo, “ConvS2S-VC: Fully convolutional sequence-to-sequence voice conversion,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1849–1863, 2020.

[22] R. Paulus, C. Xiong and R. Socher, “A deep reinforced model for abstractive summarization,” in *Proc. of 6th Int. Conf. Learning Representation*, pp. 1–12, 2018.

[23] H. Palangi, L. Deng, Y. Shen, J. Gao, X. He *et al.,* “Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval,” *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 24, no. 4, pp. 694–707, 2016.

[24] X. Duan, S. Ying, W. Yuan, H. Cheng and X. Yin, “A generative adversarial networks for log anomaly detection,” *Computer Systems Science and Engineering*, vol. 37, no. 1, pp. 135–148, 2021.

[25] P. Shaw, J. Uszkoreit and A. Vaswani, “Self-attention with relative position representations,” in *Proc. of Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, New Orleans, Louisiana, vol. 2, pp. 464–468, 2018.

[26] L. Wang, P. Yao, Y. Tao, Z. Li, W. Liu *et al.,* “A reinforced topic-aware convolutional sequence-to-sequence model for abstractive text summarization,” in *Proc. of the 27th Int. Joint Conf. on Artificial Intelligence*, AAAI Press, pp. 4453–4460, 2018.

[27] J. Xie, B. Chen, X. Gu, F. Liang and X. Xu, “Self-attention-based BiLSTM model for short text fine-grained sentiment classification,” *IEEE Access*, vol. 7, pp. 180558–180570, 2019.

[28] S. Lamsiyah, A. El Mahdaouy, B. Espinasse and S. E. A. Ouatik, “An unsupervised method for extractive multi-document summarization based on centroid approach and sentence embeddings,” *Expert Systems with Applications*, vol. 167, no. 4, pp. 114152, 2021.

[29] G. Castaneda, P. Morris and T. M. Khoshgoftaar, “Evaluation of maxout activations in deep learning across several big data domains,” *Journal of Big Data*, vol. 6, no. 1, pp. 72, 2019.

[30] A. See, P. J. Liu and C. D. Manning, “Get to the point: Summarization with pointer-generator networks,” ArXiv170404368 Cs, 2017.

[31] J. Gu, Z. Lu, H. Li and V. O. K. Li, “Incorporating copying mechanism in sequence-to-sequence learning,” in *Proc. of 54th Annual Meeting Association for Computational Linguistics*, Long Paper, vol. 3, pp. 1631–1640, 2016.

[32] T. Baumel, M. Eyal and M. Elhadad, “Query focused abstractive summarization: Incorporating query relevance, multi-document coverage, and summary length constraints into seq2seq models,” ArXiv:1801.07704, 2018.

[33] C. Y. Lin, “Rouge: A package for automatic evaluation of summaries,” in *Proc. of Association for Computational Linguistics*, Barcelona, Spain, vol. 8, pp. 74–81, 2004.