

Neural Computing and Applications

SHEG: Summarization and HEAdline Generation of News-Articles using Deep Learning --Manuscript Draft--

Manuscript Number:	NCAA-D-19-02334R2
Full Title:	SHEG: Summarization and HEAdline Generation of News-Articles using Deep Learning
Article Type:	Original Article
Keywords:	Summarisation; Headline Generation; NLP; Deep Learning
Corresponding Author:	Rajeev Kumar Singh, PhD Shiv Nadar University Dadri, Uttar Pradesh INDIA
Corresponding Author Secondary Information:	
Corresponding Author's Institution:	Shiv Nadar University
Corresponding Author's Secondary Institution:	
First Author:	Rajeev Kumar Singh, PhD
First Author Secondary Information:	
Order of Authors:	Rajeev Kumar Singh, PhD Sonia Khetarpaul Rohan Gorantla Sai Giridhar Rao Allada
Order of Authors Secondary Information:	
Funding Information:	
Abstract:	<p>The human attention span is continuously decreasing, and the amount of time a person wants to spend on reading is declining at an alarming rate. Therefore, it is imperative to provide a quick glance of important news by generating a concise summary of the prominent news article, along with the most intuitive headline in line with the summary. When humans produce summaries of documents, they not only extract phrases and concatenate them but also produce new grammatical phrases or sentences that coincide with each other and capture the most significant information of the original article. Humans have an incredible ability to create abstractions; however, automatic summarization is a challenging problem. This paper aims to develop an end-to-end methodology that can generate brief summaries and crisp headlines that can capture the attention of readers and convey a significant amount of relevant information. In this paper, we propose a novel methodology known as SHEG, which is designed as a hybrid model. It works by integrating both extractive and abstractive mechanisms using a pipelined approach to produce a concise summary which is then used for headline generation. Experiments were performed on publicly available datasets viz., CNN/Daily Mail, Gigaword, and NEWSROOM. The results obtained validate our approach and demonstrate that the proposed SHEG model is effectively producing a concise summary as well as captivating and fitting headline.</p>
Response to Reviewers:	<p>To,</p> <p>Prof. John MacIntyre & Prof. Ajay Kaul,</p> <p>We would like to thank the Editors and the Reviewers for assessing our manuscript and providing us with detailed and fruitful feedback. We would like to inform you that we have gone across all the points raised by the reviewers and prepared a detailed response as follows. We have also made the necessary changes to the manuscript. We proofread the entire document as suggested and corrected some spelling/grammatical mistakes to improve the readability.</p>

We look forward to receiving your positive feedback about the revised manuscript soon.

Kind regards,

Rajeev Kumar Singh
Sonia Khetarpaul
Rohan Gorantla
Sai Giridhar Rao Allada

Response to the Reviewer Comments (Reviewer 1)

Reviewer's Query: Well written manuscript. The proposed summarisation methodology called SHEG is the contribution and considered novel as this hybrid model integrated both extractive and abstractive mechanisms to produce a succinct news summary, thus the appropriate headline.

Our Response: We appreciate the reviewer's comments and would like to thank him/her as the quality of the article has improved substantially while addressing the pertinent points and suggestions.

Response to the Reviewer Comments (Reviewer 3)

Reviewer's Query: There is a little typo in Section 1 ("gievn" instead of "given")

Our Response: We have now corrected it and spell checked the entire document again.

Reviewer's Query: Decoder inputs are referenced as 'yt' in the Figure but as 'it' in the text. Please use a unique notation.

Our Response: It was an oversight from our side. The correction has been done by relabelling the image.

Reviewer's Query: Where do the decoder inputs come from? Are there any extra inputs to the decoder besides the encoder hidden states?

Our Response: The decoder input comes from the encoder hidden states as well as the previous word of the reference summary during training. In Fig. 2 the decoder input is labeled as i_1, i_2, \dots, i_t .

The following line has been added in the article to give more clarity to the reader. "it is the previous word of the reference summary during training while during the testing phase the previous word emitted by the decoder is considered."

Reviewer's Query: Your text mentions decoder hidden states (hd_1, hd_2, \dots, hd_n), but it seem that these values are not used in any subsequent computation, is that correct?

Our Response: The decoder hidden states are not explicitly used for any subsequent computation as such but are used to pass the information along to the next decoder states. The following point has been added to the manuscript to give more clarity. "The hidden states hd_n are used to pass information to the next decoder state."

Reviewer's Query: The color/shape notation used in your figure is not clear to me. E.g, is a given shape associated to a particular architectural component type? The same about the colors, what do similar colors indicate in the figure?

Our Response: We made the required changes to the shapes and colors to the diagram to avoid confusion. With these changes, the diagram clearly depicts the contribution of individual components. Fig. 2 has been suitably updated.

Reviewer's Query: It took me a long time to figure out that $x(pg)$ and $x(1-pg)$ stand for a simple multiplication operation. It looks like $x()$ is a function. Really confusing.

Our Response: This point has been duly noted and changes have been made to the diagram to avoid confusion and improve readability.

Reviewer's Query: In Section 3.3 please explain the meaning of PER, LOC, and ORG ("person", "location" and "organization" I suppose).

Our Response: The following changes have been made in order to make it clear and improve its clarity among readers :

“Named entities help in identifying the entities in the text using entity annotations where persons are tagged as PER, locations as LOC and organizations as ORG.”

[Click here to view linked References](#)

Noname manuscript No. (will be inserted by the editor)
--

SHEG: Summarization and HEadline Generation of News-Articles using Deep Learning

Rajeev Kumar Singh · Sonia
Khetarpaul · Rohan Gorantla · Sai
Giridhar Allada

Received: date / Accepted: date

Abstract The human attention span is continuously decreasing, and the amount of time a person wants to spend on reading is declining at an alarming rate. Therefore, it is imperative to provide a quick glance of important news by generating a concise summary of the prominent news article, along with the most intuitive headline in line with the summary. When humans produce summaries of documents, they not only extract phrases and concatenate them but also produce new grammatical phrases or sentences that coincide with each other and capture the most significant information of the original article. Humans have an incredible ability to create abstractions; however, automatic summarization is a challenging problem. This paper aims to develop an end-to-end methodology that can generate brief summaries and crisp headlines that can capture the attention of readers and convey a significant amount of relevant information. In this paper, we propose a novel methodology known as SHEG, which is designed as a hybrid model. It works by integrating both extractive and abstractive mechanisms using a pipelined approach to produce a concise summary, which is then used for headline generation. Experiments were performed on publicly available datasets viz., CNN/Daily Mail, Gigaword, and NEWSROOM. The results obtained validate our approach and demonstrate that the proposed SHEG model is effectively producing a concise summary as well as a captivating and fitting headline.

Keywords Extractive summarization · Abstractive summarization · Deep Learning · Reinforcement Learning · NLP · Headline Generation

Rajeev Kumar Singh
Shiv Nadar University
E-mail: rajeev.kumar@snu.edu.in

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

1 Introduction

A precis or a summary refers to a shorter version of an article that conveys the most salient information of the entire article. Text summarization could be defined as an act of creating a miniature portrait of an article, i.e., making an article shorter while retaining the most essential parts, thus maintaining the true essence of the document. Creating concise and coherent summaries is one of the critical issues faced by the newspaper industry, as developing a short summary of a huge article is arduous and time-consuming. Summarizing a news article is a necessity for the entire newspaper industry as it is facing stiff challenges due to strong competition from digital media houses. According to a recent study [44], the annual newspaper market in the US was valued at 27 billion dollars, and it is estimated that this value is going to fall to 17 billion by the year 2025 due to the digital boom. A 2015 report by Microsoft [36] discovered that the the average attention span dropped to about 8.5 seconds from 12 seconds. We are surrounded by technology everywhere due to the ever-growing use of smartphones, which has lead to a paradigm shift in the way we consume news. According to [47], about 80 percent of the readers never make it past the headline and the traffic of the website can vary as much as 500 percent depending on the headline. The process of headline generation suddenly seems more important for traditional news houses in order to get people to read their articles. Taking all these factors into account, traditional news houses have all gone online with content that fits on a mobile screen. It is important to have a news summary that is short, crisp, relevant and unbiased, which can also adapt to the changing landscape of reading habits.

Text Summarization could be broadly categorized into two major categories viz., Extractive and Abstractive summarization. Extractive summarization techniques are aimed at selecting salient phrases or sentences from the full text verbatim, while abstractive summarization techniques are aimed at paraphrasing with new words or phrases using the information present in the article. Text summarization has been widely studied over the last two decades and most of the early papers in this domain used extractive techniques. Extractive Summarization can be broadly categorized as graph based approaches [16], greedy approaches [5], constraint optimization approaches [34] and Deep Learning (DL) based approaches [38, 50, 58]. Recent research has emphasized more on abstractive techniques [17, 39, 45, 49]. The enormous amount of research in DL and Reinforcement Learning (RL) algorithms have broadened the scope to extend and apply these techniques into the summarization field [6, 40]. Abstractive techniques experiences slow as well as incorrect encoding of lengthy articles, requiring the attention system to look at all encoded phrases to decode each summary. Though abstractive methods have been able to produce summaries with elevated ROUGE (Recall-Oriented Understudy for Gisting Evaluation which is further discussed in section 4.2) score still they suffer from inaccurate reproduction of factual information as well as an inability to handle words out-of-vocabulary (OOV).

1 In this paper, we propose SHEG, a summary headline generator that pro-
2 duces both a precise summary and headline of the news article. It combines the
3 power of both extractive and abstractive techniques making the model supe-
4 rior on the ROUGE metric, thus promising to produce better summaries. The
5 proposed hybrid model comprises of an extractive mechanism to identify key
6 sentences or phrases from the article and a reinforced abstractive mechanism
7 which uses the key sentences/phrases produced by an extractive mechanism
8 to form a concise summary. The proposed model can handle OOV words,
9 generate appropriate factual information, avoid redundancy, fix language flu-
10 ency issues, and is faster too. The major contributions of the paper are given
11 below: (i) A hybrid approach of combining the extractive and reinforced ab-
12 stractive mechanisms to produce a summary. (ii) Use of Convolutional Neural
13 Networks (CNNs) and Recurrent Neural Networks (RNNs) to achieve word-
14 level and sentence level-attention in the extractive mechanism. (iii) Use of a
15 novel Controlled Actor-Critic (CAC) model for training the pointer-generator
16 network to strike a balance between variance and bias in the reinforced ab-
17 stractive mechanism. (iv) Use of sequence prediction in order to produce a
18 headline from the summary produced using our proposed model.

19 Section 2 provides an overview of the existing extractive and abstractive
20 summarization techniques. The proposed methodology (SHEG) is given in
21 detail in section 3. The proposed section 4 on Data and Experiments presents
22 the details of the datasets and the experimental strategies employed. Section
23 5 provides a precise analysis of the results obtained, along with a comparative
24 study of other techniques. A brief outline of the current work, as well as its
25 future scope, is given in section 6.

28 2 Related Work

29 Text summarization systems are designed to produce crisp and concise sum-
30 maries by compressing data. Humans generally paraphrase a large text by
31 understanding the key sentences and present a shorter version of the same
32 text, thus making it abstractive in nature. Researchers have been grappling
33 with the summarization problem over the past two decades, and most of the
34 earlier proposed solutions are extractive in nature. Some of the earlier liter-
35 ature like [11, 18, 24, 28] simply extracted important phrases/words but could
36 not obtain coherent paraphrases. In one of the earliest works of extractive
37 summarization, [28] had employed noisy channel framework and decision trees
38 for sentence compression. Later, [15] developed a technique that first extracts
39 verbs and nouns from the news article and then uses an iterative shortening
40 algorithm to compress the article. The latest work in extractive summariza-
41 tion is given by [33] that used [14] to design a technique known as BERTSUM,
42 which is based on a flat architecture with inter-sentence transformer layers. In
43 [42], it was stated that abstractive summarization had not progressed beyond
44 the proof-of-concept stage and remained a researcher’s dream. The emergence
45 of machine learning and deep learning techniques have opened the doors for
46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65

1 abstractive summarization research. Before the advent of neural network tech-
2 niques, abstractive summarization got less attention than extractive summa-
3 rization, but the work given in [24] performed various cut and paste operations
4 like reducing sentences, combining sentences, syntactic transformation, and
5 lexical paraphrasing for creating summaries. In [19], a graph-based approach
6 was used for the abstractive summarization of highly redundant opinions using
7 the Opinosis framework. Cheung *et al.* [8] performed sentence enhancement
8 using dependency trees. A few other noble works within the domain of ab-
9 stractive summarization incorporate traditional phrase-table based machine
10 translation approaches [4], compression utilizing weighted tree-transformation
11 rules [12] and quasi-synchronous grammar-based approaches [57].

13 In the recent past, neural network models that map an input sequence to
14 output sequence, called sequence-to-sequence models, have been effective in
15 numerous natural language tasks. One such work by [3] used the attentional
16 RNN encoder-decoder model for machine translation and achieved state-of-
17 the-art performance with a BLEU(Bilingual Evaluation Understudy) score of
18 about 28.45 . The work proposed by [45] was the first to perform abstractive
19 summarization using an encoder-decoder neural network on DUC-2004 and
20 Gigaword datasets. Later there have been research works that used attention
21 mechanism as a core idea and also augmented it with recurrent decoder [9], hi-
22 erarchical networks [39] and autoencoders [35] thereby improving performance.
23 The work presented in [39] uses a bidirectional GRU(Gated recurrent unit)
24 encoder and unidirectional GRU decoder architecture along with a pointer
25 mechanism inspired from [55]. This architecture could handle the OOV words
26 and also uses hierarchical abstraction in order to retrieve the most impor-
27 tant sentences by using an extra layer abstraction for sentence-level attention.
28 The problem of sentence repetition was addressed in [49], where a bidirec-
29 tional LSTM encoder-decoder along with a pointer network similar to that
30 used in [39] along with a coverage mechanism was used in order to address the
31 repetition problem. One of the most recent work on abstractive summariza-
32 tion by [20] uses a bottom-up attention mechanism to constrain the model to
33 likely phrases and thereby enhancing the ability to compress text, while still
34 generating fluent summaries. The authors of [39] who proposed an abstractive
35 technique also developed a neural extractive approach [38], that uses hierarchi-
36 cal RNNs to pick key sentences, and it significantly outperforms abstractive
37 result regarding the ROUGE metric. We have taken inspiration from [38] in
38 creating the quality factor and the positional impact discussed in section 3.1.
39 The quality factor considers saliency and redundancy for sentence selection
40 whereas positional impact considers the relative position of the sentence in
41 the document.

42
43 To the best of our knowledge, the earliest work to use Reinforcement Learn-
44 ing (RL) in the text summarization problem is given by [46], which is applied
45 to optimize the given score function with the given feature representation of
46 a summary. To enhance the non-differential metrics of language generation
47 and to reduce exposure bias [2,43] propose to use RL. [22] uses Q-learning
48 based RL for extractive summarization. The work done by [40] uses RL pol-
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

1 icy gradient methods for abstractive summarization. It utilizes sequence-level
2 metric rewards with curriculum learning [43] or weighted combined ML and
3 RL mixed loss [40] for language fluency and stability. More recently [6] used a
4 sentence-level policy gradient method to bridge the non-differentiable compu-
5 tation between two neural networks in a hierarchical way, while maintaining
6 language fluency.
7

8 Most of the early newspaper headline generator models were introduced
9 when the text summarization was synonymous with extractive summariza-
10 tion. Extractive summarization could not be used for headline generation as
11 it cannot produce summaries, which were short enough for it to be a head-
12 line. Hence most of these early models relied on bag-of-words concepts where
13 they heavily relied on models based on sentence position, headline word posi-
14 tion, and text modeling, i.e., the correlation of the words in text and headline
15 based on position as seen in [32]. Zhou *et al.* [61] leveraged the power of both
16 sentence positioning and text modeling to create a headline generation model
17 with less than ten words. Dorr *et al.* [15] employed a parse and trim approach
18 for generating headlines. This work uses heuristics to preserve certain parts
19 of the story and create a headline by iteratively removing constituents from a
20 parse tree of the first sentence until a length threshold has been reached. The
21 usage of the Markov model for the purpose of headline generation could be
22 observed in [59] along with several decoding parameters like length, position,
23 and gap to make it seem more like a headline. In [13], we observe that the task
24 of headline generation was viewed as a task of compression of articles in order
25 to produce headlines and their model was trained to learn how to compress
26 articles and produce headlines.
27

28 With the development of various abstractive summarization techniques
29 over the years, researchers have now taken a new approach to Headline gener-
30 ation. We do observe this in [53], where important sentences are first selected
31 using sentence based extractive mechanisms and then an abstractive summa-
32 rization based RNN encoder-decoder model is used to produce a headline. Sim-
33 ilar DL based approaches could be observed in [1] where minimum risk training
34 was introduced in order to minimize loss over the training data. Takase *et al.*
35 [52] used an attention-based approach to headline generation and is regarded
36 as an extension to [45]. This model uses a similar encoder-decoder mecha-
37 nism, but on Abstract Meaning representation, i.e., the predicate-argument
38 structures and named entities enhance the summaries that are produced.
39
40

41 **3 SHEG: Our Proposed Methodology**

42
43
44 In this section, the proposed SHEG model is described in detail. SHEG lever-
45 ages the power of both extractive and abstractive techniques for producing
46 a concise summary. Hybrid summary is then used to produce most suitable
47 headline which goes well with the generated summary. We begin by extract-
48 ing salient sentences from an extensive article using our proposed mechanism
49 described in section 3.1 followed by abstractive summarization of these key
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

sentences with the help of the reinforced abstractive mechanism given in section 3.2. We then utilize our approach to generate headlines in section 3.3.

3.1 Extractive Mechanism

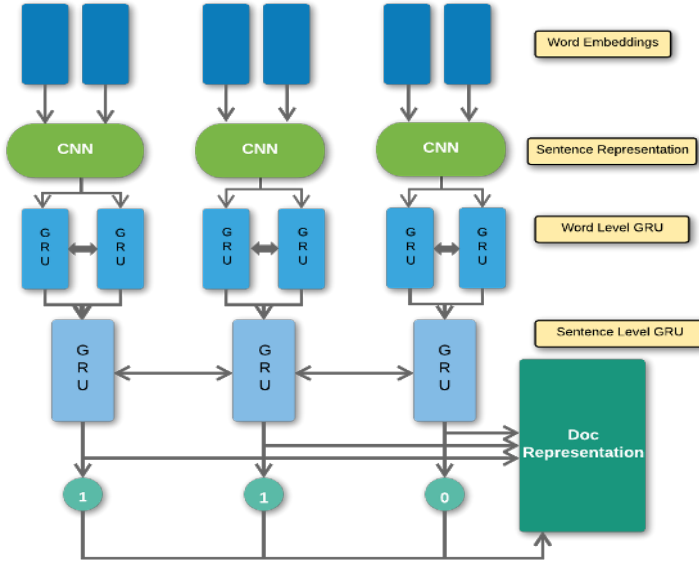


Fig. 1 Proposed Extractive Mechanism Architecture

The proposed extractive mechanism is inspired from Nallapati *et al.* [38] and is illustrated in Fig. 1. The main task of the extractive mechanism in our approach is to obtain sentence-level attention, i.e., to extract important sentences from a large article. The words in each sentence are represented in a distributed vector form using a learned embedding matrix W_{emb} . Word2vec [41] algorithm is employed to construct word embeddings. It is a two-layer neural net that vectorizes words to process text, and takes text as input and produces a collection of vectors as outputs, i.e., feature vectors representing words in the body. These embeddings will contain information such as the context of a word in a document and capture the semantic and syntactic similarity between the words. CNNs, as proposed by [26], are used to determine the representation of each sentence. For this task, word embeddings are passed through the 1-D convolution kernels with different window sizes of 3, 4, 5 to comprehend the dependencies of nearby words. The convolutional layer is followed by three layers viz., Rectified Linear Unit (ReLU), non-linear activation, and max-over-time pooling layers. On concatenating the activation outputs of

various filter window sizes, a convolutional representation of the sentence is obtained. As shown in Fig. 1, CNN is followed by a two-layer bi-directional GRU [10]. The first GRU layer operates at word level within each sentence. It computes the hidden state representations based on the previous hidden state and the current word embeddings. The second layer works over sentences by taking hidden states of the first bi-directional GRU layer as input. The hidden states obtained from the second bi-directional GRU layer provide encodings of sentences in the document. The representation of article can be mathematically modeled as a non-linear transformation of the average pooling of the concatenated hidden states of the bi-directional sentence-level GRU as shown below [38].

$$A = \tanh\left(W_a \frac{1}{N_a} \sum_{k=1}^{N_a} [h_k^f, h_k^b] + b\right) \quad (1)$$

where h_k^f and h_k^b are the hidden states corresponding to the k^{th} sentence of the forward and backward sentence-level RNNs respectively, N_a is the number of sentences in the document and $[\cdot, \cdot]$ represents vector concatenation. After this step, sentence level classification takes place i.e., each sentence is checked whether it belongs to the summary or not based on the quality of the sentence. It is represented by the equation below [38].

$$Q(h_k, m_k, A) = W_c h_k + h_k^T W_s A - h_k^T W_r \tanh(m_k) \quad (2)$$

The quality factor (Q) takes into account the content of the sentence which is signified by $W_c h_k$, the saliency of the sentence denoted by $h_k^T W_s A$ and the redundancy of the sentence with regard to current state of summary is represented by $h_k^T W_r \tanh(m_k)$. Here W_c , W_s and W_r refer to the corresponding weights. m_k refers to the dynamic representation of the k^{th} sentence summary. We also consider the position of each sentence k in a summary which is known as Positional Impact (PI). PI depends on the absolute as well as relative positional embeddings p^α and p^γ . PI is calculated using the equation below [38].

$$PI = W_\alpha p_k^\alpha + W_\gamma p_k^\gamma \quad (3)$$

W_α and W_γ gives the corresponding weights for both absolute and relative positions. Finally the probability of the sentence selection using various factors viz. quality factor Q , PI and bias b is computed using the equation given below

$$P(y_k = 1 | h_k, m_k, A) = \sigma(Q(h_k, m_k, A) + PI + b) \quad (4)$$

where y_k is the binary value representation of whether the k^{th} sentence belongs to the summary or not and h_k refers to the concatenated hidden states at the k^{th} time step. Hence the dynamic representation of the summary at the k^{th} sentence position, could be represented by

$$m_k = \sum_{j=1}^{k-1} h_j P(y_j = 1 | h_j, m_j, A) \quad (5)$$

Although we have devised sentence level extraction as a classification problem but many of the existing summarization datasets are end to end document-summary pairs that lack saliency labels for each sentence. In order to counter this, we use a similarity method to provide a proxy target label for the extractor i.e., the most similar article sentence \mathbf{A}_{k_t} is found for each ground truth sentence and it is represented by:

$$k_t = \operatorname{argmax}_i(\operatorname{ROUGE} - L_{\operatorname{recall}}(\mathbf{A}_i, \mathbf{g}_t)) \quad (6)$$

where \mathbf{g}_t refers to the ground truth summary. The task of maximizing ROUGE score by finding globally optimal subset of sentences is computationally expensive therefore we opted for a greedy approach. In this method, one sentence is selected at a time, such that the ROUGE score of the current set of sentences is maximized with respect to the reference summary. This process is stopped when there is no further improvement of ROUGE score on addition of remaining sentences to the current summary set.

The negative log-likelihood loss is thus reduced and can be represented by

$$\begin{aligned} \ell(\mathbf{W}, \mathbf{b}) = & - \sum_{a=1}^N \sum_{k=1}^{N_a} (y_k^a \log P(y_k^a = 1 | \mathbf{h}_k^a, \mathbf{m}_k^a, \mathbf{A}_a)) \\ & + (1 - y_k^a) \log (1 - P(y_k^a = 1 | \mathbf{h}_k^a, \mathbf{m}_k^a, \mathbf{A}_a)) \end{aligned} \quad (7)$$

3.2 Reinforced Abstractive Mechanism

In the previous section, we discussed about the proposed extractive mechanism for identifying the most important and salient sentences by employing a sentence level attention architecture as shown in Fig. 1. In this section a novel reinforced abstractive mechanism is proposed which works on the extracted key sentences to create a concise summary. The proposed reinforced abstractive mechanism takes inspiration from pointer-generator network given by [49] for copying words in the article as well as generating new words from a fixed vocabulary simultaneously. The proposed architecture is shown in Fig. 2.

We use a bidirectional LSTM encoder where the hidden states are denoted by \mathbf{h}_e^n and unidirectional LSTM decoder whose hidden states are denoted by \mathbf{h}_d^n for the n^{th} word, along with a pointer network inspired from [55] in order to deal with OOV words. It tends to replace these OOV words by pointing to words in the original text itself. This pointer generator network could either choose the next word in the summary from the existing document by pointing or by generating a fixed word from the existing vocabulary. The probability of generation at timestep t is given by [49].

$$p_g = \sigma(\mathbf{w}_h^T \mathbf{h}_t^c + \mathbf{w}_d^T \mathbf{d}_t + \mathbf{W}_i^T \mathbf{i}_t + \mathbf{b}_p) \quad (8)$$

where \mathbf{w}_h^T , \mathbf{w}_d^T , \mathbf{w}_i^T and \mathbf{b}_p are learnable parameters. \mathbf{h}_t^c refers to the context vector at timestep t , \mathbf{d}_t refers to the decoder state and \mathbf{i}_t refers to decoder input. \mathbf{i}_t is the previous word of the reference summary during training while

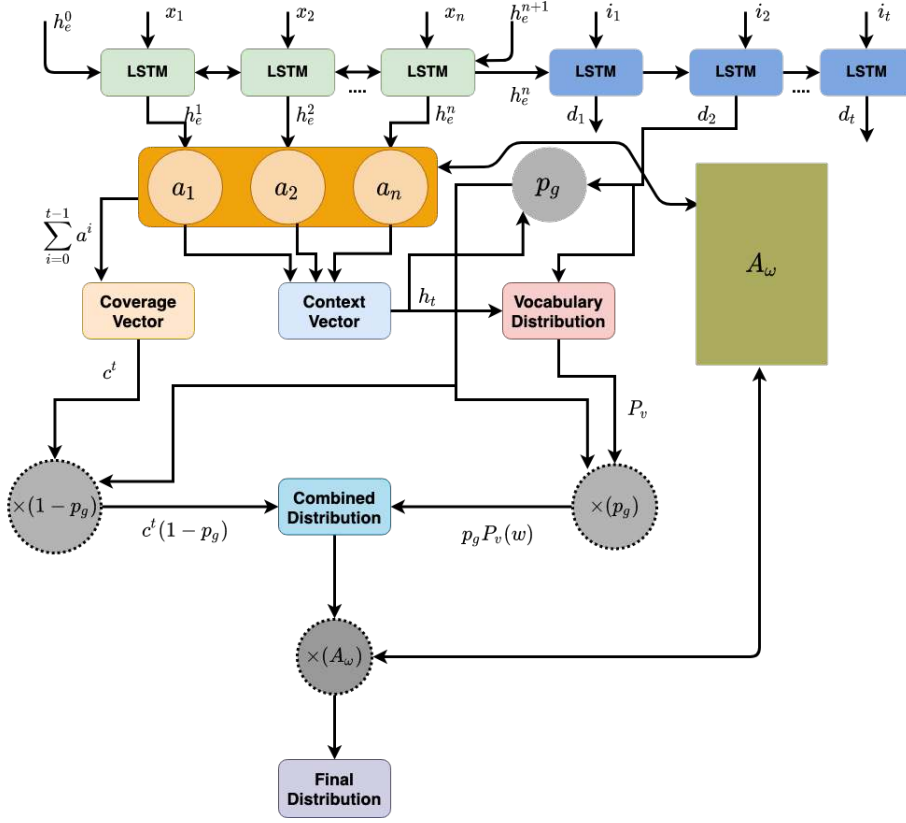


Fig. 2 Reinforced Abstractive Mechanism Architecture

during the testing phase the previous word emitted by the decoder is considered. The hidden states h_d^n are used to pass information to the next decoder state. The context vector h_t can be represented by the following equation.

$$h_t^c = \sum_n a_n^t h_e^n \quad (9)$$

where a_n^t refers to the attention distribution at timestep t . The context vector, for that particular time can be seen as a set size depiction of what was read from the source and along with the decoder state d_t is considered to form the vocabulary distribution P_v given by the below equation [55].

$$P_v = \text{softmax}(W_a(W_b[d_t, h_t] + b_b) + b_a) \quad (10)$$

where W_a , W_b , b_a and b_b are learnable parameters.

The probability of whether the word w would be decoded next could be identified from the probability distribution function as given below,

$$P(w) = p_g P_v(w) + (1 - p_g) \sum_{n: w_n = w} (a_n^t) \quad (11)$$

Coverage mechanism featured in [49] has been employed to counter the abstractive mechanism from attending the same location repeatedly. It is used for the calculation of future attention distributions in order to prevent the repetition of phrases. A coverage vector \mathbf{c}^t is represented by the sum of all the previous attention distributions and can be denoted as follows [49]

$$\mathbf{c}^t = \sum_{i=0}^{t-1} \mathbf{a}^i \quad (12)$$

Recently the work done by [40] uses Policy Gradient (PG) reinforcement learning (RL) algorithm given by [56] for abstractive summarization, however this method suffers from high variance. Another notable work on abstractive summarization by [30] used Actor-Critic (AC) model for training, AC models generally have low variance owing to batch training and the use of critics as the baseline reward. To overcome the drawbacks of PG and AC models, we propose a novel RL based method for training the pointer-generator network called Controlled Actor-Critic (CAC) Model. Generally, in RL algorithms an agent takes action depending on a particular policy π . Some of the terminologies used are explained below - \mathbf{S} is the set of states, \mathbf{A} is the set of actions, $\mathbf{R} : \mathbf{S} \times \mathbf{A} \rightarrow \mathbb{R}$ is reward function, $\gamma \in [0, 1]$ is the discount factor and H is the horizon. Suppose the current state at abstraction step K is \mathbf{s}_K , the agent picks an action $\hat{y}_K \in \mathbf{A}$ based on a stochastic policy $\pi(\hat{y}_K | \mathbf{s}_K) : \mathbf{S} \times \mathbf{A} \rightarrow \mathbb{R}$ and receives reward r_K for action \hat{y}_K . The goal of the agent is to maximize the expected discounted reward R_K [51]

$$R_K = \mathbb{E}_\pi \left[\sum_{v=0}^H \gamma^v r_v \right] \quad (13)$$

where the discounting factor γ keeps a balance between immediate and future rewards. The value function V intuitively measures how useful the model might be when it is in a particular state \mathbf{s} . However, being in such a state, the Q function measures the importance of selecting a particular action. Value function $V_\pi(\mathbf{s}_K)$ is defined only on the states and $Q_\pi(\mathbf{s}_K, y_K)$ which is based on both the states as well as actions. The mathematical expression for both the functions are given below [51].

$$Q_\pi(\mathbf{s}_K, y_K) = \mathbb{E}[r_K | \mathbf{s} = \mathbf{s}_K, y = y_K] \quad (14)$$

$$V_\pi(\mathbf{s}_K) = \mathbb{E}_{y \sim \pi(\mathbf{s})}[Q_\pi(\mathbf{s}_K, y = y_K)] \quad (15)$$

Using equations 14 and 15, a new function called advantage, A_π is defined as given below [25].

$$A_\pi(\mathbf{s}_K, y_K) = Q_\pi(\mathbf{s}_K, y_K) - V_\pi(\mathbf{s}_K) \quad (16)$$

We consider pointer-generator network as an RL agent, whose action denotes choosing the next token for the summary and its reward function is based on ROUGE score. The output state at each time step of the pointer-generator network is used as current state of the CAC model to calculate Q , V and A_π function. A critic network with trainable parameters ω is used to approximate the state value function $V_\pi(\mathbf{s}; \omega)$. In order to strike a balance between bias

and variance, a control advantage function is defined similar to [48] as given below

$$A_\omega(s_\kappa, y_\kappa) = \sum_{i=\kappa} (\gamma\lambda)^{i-\kappa} (r(s_i, y_i) + \gamma V_\omega(s_{i+1}) - V_\omega(s_i)) \quad (17)$$

Where λ keeps check on the bias and variance trade-off, wherein large values of λ yield larger variance and lower bias, while small values of λ works in the reverse way. The actor uses the value estimates V_ω to calculate the controlled advantage approximate A_ω and updates the loss according to the following equation [25].

$$\mathcal{L}_\omega = \frac{1}{N} \sum_{i=1}^N \sum_{\kappa} \log \pi_\omega(\hat{y}_{i,\kappa} | \hat{y}_{i,\kappa-1}, s_{i,\kappa} c_{i,\kappa-1}) A_\omega(s_{i,\kappa}, y_{i,\kappa}) \quad (18)$$

3.3 Headline Generation

Our headline generation technique is aimed at generating an appropriate and most informative headline which is in line with the concise summary produced by the hybrid summarizer. One of the key challenges faced while creating the headline is about choosing a limited number of words to convey the main points of an article. The proposed technique models the headline generation as a sequence prediction job. We use one of the well-known sequence model known as Conditional Random Fields (CRFs) proposed by Lafferty *et al.*, [29]. Let $X = \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$ be a finite set of possible observations, and let $Y = \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n$ be a finite set of possible categories that each observation could belong to. In general, statistical sequence models approximate probability distribution P with parameters Φ capable of predicting the probability $P(\mathbf{y}|\mathbf{x}; \Phi)$ for any sequence of n observations $\mathbf{x} \in X_n$ and any sequence of assigned categories per observation $\mathbf{y} \in Y_n$. The probability of the sequence can be modeled by the CRF using the following equation [13].

$$P(\mathbf{y}|\mathbf{x}; \Phi) = \frac{e^{\mathbf{w} \cdot \Theta(\mathbf{x}, \mathbf{y})}}{N(\mathbf{x})} \quad (19)$$

where $\Phi = \mathbf{w}$ and $\mathbf{w} \in \mathbb{R}^p$ is a weight vector, $N(\mathbf{x})$ is the normalisation function, $\Theta : X^n \times Y^n \rightarrow \mathbb{R}^p$ is a function which denotes global features with p dimensions. Let μ be a state space, μ_0 be a fixed initial empty state, $f : X^* \times \mathbb{N}^+ \times Y \times Y \rightarrow \mathbb{R}^p$ represents local feature function and let function $g : \mu \times X^* \times \mathbb{N}^+ \times Y \rightarrow \mu$ model state transitions. The global feature function which transforms the model into a log-linear model in the feature space can then be defined using the following equation [37].

$$\Theta(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^n f(\mathbf{x}, i, \mu_{i-1}, y_i) = g(\mu_{i-1}, \mathbf{x}, i, y_i) \quad (20)$$

The best category can then be computed using the equation given below [13].

$$\hat{y} = \arg \max_{y \in Y^n} w. [\sum_{i=1}^n f(x, i, \mu_{i-1}, y_i)] \quad (21)$$

The obtained summary of the news article is seen as a series of observations, where each observation is a possible token in the article summary and each observation can be assigned to either of the two categories: present in the headline, or not present in the headline. The main objective of the local feature function is to return a vector that describes in an euclidean space the outcome of placing, or not placing, token \mathbf{x}_i in the headline, given that the previous words are chosen from the state μ_{i-1} . The feature vector consists of various indicators based on many parameters such as *named entities*, *dependency features*, *language model features* and the *headline length*. *Named entities* help in identifying the entities in the text using entity annotations where persons are tagged as PER, locations as LOC and organisations as ORG. *Dependency features* inform the model about syntactical dependencies among the tokens placed on the headline with the help of a dependency tree. Dependency tree is built with the help of a parse tree built using a parser present in Stanford coreNLP library. *Language model features* examines the grammaticality of the headline generated using conditional bigram probability. It is conditional on the previous token being inserted in the headline of the current token and the trigram probability of the PoS tag. The *headline length* ensures length of generated headline (an ideal headline must be in 8-10 words according to previous studies [4]) by checking the length of the headline at every step and checking the presence of the token in the headline. The feature vector takes into consideration various parameters as described and will only fire if the token \mathbf{x}_i is planted in the headline i.e., $y_i = \mathbf{1}$. In order to find the global optimum in polynomial time, a dynamic programming approach is used similar to [13]. The global optimum is found by producing all possible states and using the state that from the initial state, generates the maximum score. The global feature function θ takes the document and bitmap as input, and generates a vector that describes the headline in an abstract feature space. We define the feature function in such a way that it focuses on evaluating how a series of tokens that comprise a headline relate to each other and to the document as a whole. We describe the algorithm below, for finding the top-scoring state μ^* which in turn helps in finding the global optimum of our model's objective function. We compute iteratively $\Omega(i, l)$, which returns the set of all reachable states that correspond to headlines having token-length l and ending with token \mathbf{x}_i . The algorithm also computes $\Delta(\mu)$, which returns the maximum score that can be obtained by following a chain of state sequences which begins on μ_0 and ends in the provided state.

Algorithm 1: *Headline Generation Algorithm*

```

1  Assigning Variables:
2  1  $M \leftarrow$  Maximum number of tokens in headline
3  2  $k \leftarrow$  Number of tokens in the article
4  3  $\mathbf{x} \leftarrow$  list of  $k$  tokens in the article
5  4  $f \leftarrow$  Local feature function
6  5  $g \leftarrow$  State transition function
7  6  $\mu_0 \leftarrow$  Init State
8  7  $\mathbf{w} \leftarrow$  Weight vector
9  Initializations:
10 8  $\Omega \leftarrow$  Set of States  $[l+1][M+1]$ 
11 9  $\Delta \leftarrow$  List of type float to save scores of each state
12 10  $\mu^* \leftarrow \mu_0$ 
13 11  $\Delta(\mu_0) \leftarrow 0$ 
14 12 for  $j = 1, 2, \dots, k$  do
15 13    $\Omega(i, 0) \leftarrow \{\mu_0\}$ 
16  Finding top-scoring state:
17 14 while  $p = 1, 2, \dots, M$  do
18 15   while  $i = p, \dots, k$  do
19 16    while  $q = p - 1, \dots, i - 1$  do
20 17     for  $r$  in  $\Omega(q, p - 1)$  do
21 18       $\mu \leftarrow g(r, \mathbf{x}, i, 1)$ 
22 19       $\mu_{score} \leftarrow \Delta(r) + \mathbf{w} \cdot f(\mathbf{x}, i, r, 1)$ 
23 20       $\Omega(i, r) \leftarrow \Omega(i, r) \cup \{\mu\}$ 
24 21       $\Delta(\mu) \leftarrow \max(\Delta(\mu), \mu_{score})$ 
25 22      if  $\Delta(\mu) > \Delta(\mu^*)$  then
26 23        $\mu^* \leftarrow \mu$ 

```

In order to perform learning, we move through the training data iteratively and carry out the weight updates at each stage using the equation given below [13]:

$$\mathbf{w}^* \leftarrow \mathbf{w} + \rho \times (\mathbf{c} - \hat{\mathbf{d}}), \hat{\mathbf{d}} = \Theta(\mathbf{x}, \hat{\mathbf{y}}) \quad (22)$$

where $\hat{\mathbf{y}}$ is calculated using the equation (21). \mathbf{c} represents the headline's h relationship to document \mathbf{x} and is defined as $\mathbf{c} = \Theta(\mathbf{e}, \mathbf{o})$ where \mathbf{e} is the concatenation of h and \mathbf{x} , and \mathbf{o} is the bitmap for selecting the headline tokens. $\rho \in \mathbb{R}$ is the learning factor and is calculated as given below [13]:

$$\rho = \frac{1 - \mathbf{w} \cdot (\mathbf{c} - \hat{\mathbf{d}})}{\|\mathbf{c} - \hat{\mathbf{d}}\|^2} \quad (23)$$

4 Data and Experiments

4.1 Data

“SHEG: Summarization and HEadline Generation of News-Articles using Deep Learning” as the name suggests is a summary as well as a headline generator hence the evaluation of our model has been done on three important outcomes

viz., produced extractive summary, produced abstractive summary and generated headline. To evaluate this proposed model, we have used the CNN/Daily Mail dataset for the summary produced and Gigaword dataset for the headline produced. We have also made use of the Cornell NEWSROOM Dataset for the end-to-end testing of our model as it is the only dataset currently available that consists of both summary and headline for testing. The basic properties of the datasets used are described below in Table 1.

Table 1 Properties of all Datasets used

Dataset	No. of News articles	Average article size (in words)	Average summary size (in words)
CNN/Daily Mail	287k	781	56
Gigaword	10 Million	31	8.3
NEWSROOM	1.3 Million	658.6	26.7

4.1.1 CNN/Daily Mail Dataset

The CNN/Daily Mail dataset is part of the DeepMind Q&A Dataset which was created in 2015 and consists of around 287K news articles having 2 to 4 summary sentences for each news article. This dataset contains online news articles (781 words on an average) paired with multi-sentence summaries (3.75 sentences or 56 tokens on an average). The processed version of the dataset contains about 287,226 training pairs, 13,368 validation pairs and 11,490 testing pairs.

4.1.2 Gigaword Dataset

For headline generation we use the Gigaword dataset which has been preprocessed using Stanford CoreNLP. It consists of nearly 10 million news articles from 7 news outlets. The complete training vocabulary of Gigaword consists of about 119 million word tokens and 110K unique word types with an average sentence size of about 31.3 words however we consider only a subset for training and testing the headline generator of the model i.e., the headline training input consists of about of 31 million tokens and 69K word types with the average title length of 8.3 words which as mentioned in the above section is ideal for the length of the headline we intend to produce. We have set aside 1500 headline-article pairs of the dataset solely for the purpose of testing.

4.1.3 NEWSROOM Dataset

For the end to end testing we use the Cornell NEWSROOM dataset. This dataset is the latest in the field of news article summarization and boasts of having about 1.3 million articles from 38 major news publications over the span

of 20 years as mentioned in [21]. This dataset is in the form of a compressed json line format from which we extract the text i.e., the article in this context, summary and the headline . It also has test data spanning 995k articles. The mean article size is about 658.6 words and mean summary length is about 26.7 words which should be perfect for the headline generator. This dataset consists of summaries from different extraction strategies i.e., Abstractive, Extractive and Mixed of which we have used just used the Abstractive summaries. We have used this dataset primarily for the testing purpose as it consists of both the article summary as well as the headline. This dataset has supported the evaluation of SHEG on all it's parameters in a useful way.

4.2 Training and Evaluation

In order to quantify the results, the ROUGE scoring metric has been used to check how close our machine generated summary is to the human written summary given in the dataset. ROUGE stands for Recall-Oriented Understudy for Gisting Evaluation [31] i.e., it is an evaluation metric for rating machine generated summaries compared to those written by humans. We consider the ROUGE-1 (R-1), ROUGE-2 (R-2) and ROUGE-L (R-L) scores to rate our model and use these scores to compare them with other models which have been proposed in the past.

$$ROUGE - N = \frac{\sum_{S(rf)gram_n} \sum_S Count_{match}(gram_n)}{\sum_{S(rf)gram_n} \sum_S Count(gram_n)} \quad (24)$$

where $\sum_{S(rf)gram_n} \sum_S Count_{match}(gram_n)$ refers to the total sum of n-grams that match between the reference summary and produced summary and $\sum_{S(rf)gram_n} \sum_S Count(gram_n)$ refers to total sum of n-grams in both the summaries. ROUGE-1 gives the overlap between the summary of the machine generated and the summary of the reference of unigrams. ROUGE-2 gives the overlap between machine generated and reference summaries of bi-grams. ROUGE-L is be defined as a measure of the longest matching sequence of Longest Common Subsequence (LCS) phrases. The ROUGE score is a fairly subjective measure. It could only be used to compare two models on a particular dataset and its measure could only tell us about the goodness of that model on that particular dataset. The ROUGE metric should not be used over different datasets as the size (total sum of n-grams) and overlap may vary from dataset to dataset.

All hyperparameters are tuned on the validation set of the original text version of CNN/Daily Mail. The extractive model uses pretrained word2vec [41] embeddings of size 100. The hidden state size is set at 200. The batch size is set to 64, and adam optimizer is employed [27] with a learning rate of 0.001 for training CNNs and GRUs. Gradient clipping with a maximum norm of 2 was used to regularize the model and an early stopping criterion was set based on the validation cost. The vocabulary size is limited to 150K and the maximum number of sentences per document is set to 100, and the maximum

1 sentence length is set to 60 words, to speed up computation. The extractive
2 mechanism model is trained until convergence. For the reinforced abstractive
3 mechanism, an encoder with 256 hidden states for both directions in the one-
4 layer LSTM, and 512 for the one-layer decoder was utilized. The embedding
5 size is set to 128. We observed that augmenting the model size or changing the
6 model to transformer leads to slight improvement in performance, but at the
7 cost of increased training time and parameters [54]. A learning rate of 0.0001
8 is used for training our RL algorithm with the help of adam optimizer.
9

10 11 **5 Results and Discussion**

12
13 In this section the results we obtained after extensively testing SHEG is dis-
14 cussed. The comparison of our model against other state of the art models in
15 the field of newspaper article summarization and headline generation is de-
16 scribed in detail. We evaluate our model on four areas , the i)Extractive Sum-
17 mary which is first produced after picking sentences based on their saliency,
18 ii)Abstractive Summary which is produced using the selected salient sentences,
19 iii)Headline Generation from the produced summary and iv)End-to-End Val-
20 idation in order to test the entirety of the model i.e both the summary and
21 headline produced.
22
23

24 25 **5.1 Extractive Summary**

26
27 SHEG as already mentioned is a hybrid summarization model hence a combi-
28 nation of extractive and abstractive summarization is used in order to attain
29 a summary. The very first step of our hybrid approach involves producing an
30 extractive summary of the article presented. The extractive layer basically be-
31 haves like a filter and selects sentences that are high on quality and takes into
32 consideration the context of the sentence, saliency and redundancy. In Table
33 2, we compare various state of the art extractive models with the proposed
34 model. We observe that the extractive model of SHEG significantly outper-
35 forms NeuralSummarizer. SHEG+conv's superior performance with respect
36 to SummaRuNner could be attributed to the to the presence of the convo-
37 lutional layer in the proposed extractive mechanism that is used for sentence
38 level representation or first level sentence encoding. The presence of this con-
39 volutional layer helps in handling sentences of variable length and is known
40 for picking prominent feautres i.e., prominent sentences. This could be ob-
41 served in Table 3 where the SHEG summarizer produces a better extractive
42 summary than SummaRuNner as it picks up the most salient information re-
43 quired. Though the SHEG summarizer outperforms several models it does not
44 outperform the current best extractive summarizer BERTSUM however this
45 could be attributed to the fact that BETRSUM is a sole extractive summa-
46 rizer which isn't the goal SHEG intends to achieve. In Table 4 we do observe
47 that our SHEG extractor is a better fit to our pipeline when compared to the
48 BERTSUM extactor in the production of the abstractive summary.
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

Table 2 Comparative study of various extractive models on the CNN/Dailymail dataset

Model	ROUGE-1	ROUGE-2	ROUGE-L
SummaRuNNer [38]	39.6	16.2	35.3
BERTSUM [33]	43.22	20.17	39.57
NeuralSummarizer [7]	35.5	14.7	32.2
SHEG(extraction)	40.23	14.6	36.6
SHEG+conv(extraction)	42.5	17.6	35.6

Table 3 Extractive Summaries

Article
london, england -lrb- reuters -rrb- – harry potter star daniel radcliffe gains access to a reported # 20 million -lrb- \$ _41.1_ million -rrb- fortune as he turns 18 on monday, but he insists the money won’t cast a spell on him . daniel radcliffe as harry potter in “ harry potter and the order of the phoenix ” to the disappointment of gossip columnists around the world, the young actor says he has no plans to _fritter_ his cash away on fast cars, drink and celebrity parties . “ i don’t plan to be one of those people who, as soon as they turn 18, suddenly buy themselves a massive sports
Reference Summary
harry potter star daniel radcliffe gets # 20m fortune as he turns 18 monday . young actor says he has no plans to _fritter_ his cash away . radcliffe ’s earnings from first five potter films have been held in trust fund
SummaRunner Summary
harry potter star daniel radcliffe gains access to a reported # 20 million -lrb- \$ _41.1_ million -rrb- fortune as he turns 18 on monday, but he insists the money won’t cast a spell on him . daniel radcliffe as harry potter in “ harry potter and the order of the phoenix ” to the disappointment of gossip columnists around the world, the young actor says he has no plans to _fritter_ his cash away on fast cars, drink and celebrity parties .
SHEG Summary
harry potter star daniel radcliffe gains access to a reported # 20 million -lrb- \$ _41.1_ million -rrb- fortune as he turns 18 on monday, but he insists the money wo n’t cast a spell on him .his agent and publicist had no comment on his plans’ radcliffe ’s earnings from the first five potter films have been held in a trust fund which he has not been able to touch.

5.2 Abstractive Summary

We perform the reinforced abstractive mechanism on the extractive summary obtained. In order to show the true importance of the extractive mechanism we compare our results on both the extracted text as well as the original groundtruth and these results are shown in Table 4. This shows criticality of the extractive mechanism in our hybrid model. We also compare our methodology with the well known pointer-generator network [49]. Our model outperformed it due to the additional CAC algorithm used for training as well as the effective extractive mechanism, which helped in reducing redundancy. It consistently gave better results as compared to [17]. We also evaluated our model with respect to other RL models in abstractive summarization. Some of these models have either suffered from high variance [40] or low variance [30], however in our CAC model we have imbibed the parameter λ which created

a significant difference by maintaining a balance between bias and variance thereby improving the performance.

Table 4 Comparitive study of various Abstractive Summarization models on the CNN/Daily Mail dataset

Model	ROUGE-1	ROUGE-2	ROUGE-L
HierAttn [39]	32.75	12.21	29.01
Pointer Generator [49]	39.53	17.28	36.38
Fan 2017 [17]	39.75	17.29	36.75
DeepRL [40]	39.87	15.82	36.90
Fast abstractive RL [6]	40.04	17.61	37.59
Unified model [23]	40.19	17.67	36.38
BERTSUM(extraction)+SHEG(abstraction)	40.26	17.71	36.44
SHEG(extraction + abstraction)	40.67	17.74	36.69

On closely observing SHEG’s summary in Table 3, we notice the line *his agent and publicist had no comment on his plan's radcliffe's earnings from the first five potter films have been held in a trust fund which he has not been able to touch. over daniel radcliffe as harry potter in "harry potter and the order of the phoenix" to the disappointment of gossip columnists around the world, the young actor says he has no plans to fritter his cash away on fast cars, drink and celebrity parties*, given by the SummaRuNNer summary as it scores the latter with a lower Sentence distribution compared to the other. We could observe that this extractive summary acts as a precursor to the abstractive summary produced which can be clearly seen in Table 5.

Table 5 Abstractive Summaries

Article
london, england -lrb- reuters -rrb- – harry potter star daniel radcliffe gains access to a reported # 20 million -lrb- \$ _41.1_ million -rrb- fortune as he turns 18 on monday, but he insists the money won't cast a spell on him . daniel radcliffe as harry potter in “ harry potter and the order of the phoenix ” to the disappointment of gossip columnists around the world, the young actor says he has no plans to _fritter_ his cash away on fast cars, drink and celebrity parties . “ i don't plan to be one of those people who, as soon as they turn 18, suddenly buy themselves a massive sports
Reference Summary
harry potter star daniel radcliffe gets # 20m fortune as he turns 18 monday . young actor says he has no plans to _fritter_ his cash away . radcliffe 's earnings from first five potter films have been held in trust fund
Fast abstractive RL Summary
harry potter in “ harry potter and the order of the phoenix . he says he has no plans to fritter his cash away on fast cars . radcliffe . he has been held in a casino, but he says .
Unified Model Summary
harry potter star gets reported #20 million fortune on Monday.18 year old says he has no plans to fritter cash on cars . Earnings held in trust.Actor keeping feet firmly on ground
SHEG Summary
harry potter daniel radcliffe to get # 20 million -lrb- as he turns 18 but insists won't cast spell.No current plans. Earnings from five potter films held in trust fund.

When we compare the results of SHEG to the unified model, we observe the line *Earnings from five potter films held in trust fund* in our summary that wasn't present in the unified model summary. This line was picked up by our model extractor as one can see and due to the reinforcement learning the model has not picked up unnecessary lines like *Actor keeping feet firmly on ground..*. Thus the results demonstrate a significant improvement in performance as compared to the existing state-of-the-art methodologies.

5.3 Headline Generation

The final step of SHEG is to produce a headline over the abstractive summary produced in the previous step. For this task, we have employed sequence prediction to produce the intended headline. Though earlier models like [4, 13, 53] have produced headlines using newspaper articles, our model stands out due to the fact that we use an abstractive summary of the original news article for producing a headline. The local feature function as discussed in section 3.2 iterates through each token in the produced abstractive summary and chooses tokens that could be present in the produced headline. It takes named entities, i.e., the names of people, locations, etc. into consideration while assigning membership to a particular token in the headline. It also takes into consideration syntactical dependencies i.e., the dependency among tokens to choose the next token depending upon the previous state. The function also limits the length of the headline to 8-10 words and ensures grammatical saliency. All possible states are then produced, and each state is scored using the global feature function. The state with the best score is then deemed as the headline. As seen in Table 6 SHEG tends to compete with other neural network based

Table 6 Comparative study of various Headline Generation models on the Gigaword dataset

Model	ROUGE-1	ROUGE-2	ROUGE-L
Summ-hieratt [1]	29.6	8.17	26.05
HEADS [13]	31.3	9.1	26.20
ABS+AMR [52]	31.64	12.94	28.54
SHEG	31.82	13.2	28.80

headline models like [52] and [1]. These models usually work on basic encoder-decoder based frameworks similar to other summarization models hence they do produce good results on the ROUGE metric however SHEG tends to produce a superior ROUGE score since the sequence prediction model usually condenses the article in order to produce a headline. Since the article is already condensed due to the abstractive summarization in the previous step it produces better results.

5.4 End-to-End Validation

The SHEG model is finally tested for an end-to-end validation using the Cornell NEWSROOM dataset. We use the title and reference summary given in the dataset to evaluate the headline and summary produced by SHEG based on its ROUGE score. We observe a ROUGE-1 score of about 28.2, ROUGE-2 score of 14.7 and ROUGE-L of about 25.9 for the summaries produced. In the case of the headline generated the ROUGE score might be considerably lower due to the fact that SHEG considers the ideal length of a headline to be about 8-10 words. An actual headline might have quite fewer words and the actual headline need not be a necessary condensation of the news article therefore it could be rephrased using other terms as well. This phenomenon can be observed in Tables 7 and 8.

Table 7 NEWSROOM SHEG Summary

Headline
Fountain of youth: Could this beer make you more beautiful?
Generated Headline
Collagen Beer makes you look youthful and beautiful
Article
It's not just Japan, either. In Malaysia, the Bone & Pot steamboat restaurant (Yau Guat Hei), sells a signature dish called "Collagen Soup," using cubes of collagen jelly shipped from Japan, which the business claims is perfect for "beauty & confidence." Collagen is a fibrous protein that makes up part of the connective tissues in our bodies. The body needs collagen as it helps with skin elasticity and for "looking youthful." However, as we get older, the production of collagen in the body slows down. Read MoreJohnnie Walker reinvents the "glass" But can ingesting collagen really be that good for you? In 2006, Naoya Matsuda of the Hiroasaki University School of Medicine and others published a paper on the "effects of ingestion of collagen peptide," which suggested that ingestion of collagen peptide could improve
Reference Summary
Suntory, has launched "Precious," a light beer targeted predominantly at women. Yet this drink has a secret ingredient: collagen.
SHEG Summary
Collagen increases skin elasticity and makes you youthful. Beer has secret ingredient targeted at women

Table 8 Results of the SHEG model on the NEWSROOM dataset

Abstractive Summary Result			
Model	ROUGE-1	ROUGE-2	ROUGE-L
SHEG(extraction + abstraction)	28.2	14.7	25.9
Generated Headline Result			
SHEG	13.81	7.94	11.42

6 Conclusion

In this paper we have proposed a novel methodology called SHEG, to summarize news articles and produce a suitable and crisp headline. The proposed model picks the salient phrases through its extractive mechanism and then combines the power of a pointer-generator network and CAC in order to form an abstractive summary which is then used to form a headline that would convey relevant information and would be interesting enough to seize the attention of a reader. SHEG has outperformed state of the art models and is one of its kind model that produces both an abstractive summary and a related headline. This model was trained, tested and validated by using the CNN/Daily Mail, Gigaword and NEWSROOM datasets respectively. We firmly believe that this proposed methodology can be applied to other summarization tasks in various fields like legal and medical domains, giving it the possibility of wide applicability in the field of Natural Language Processing and Information Retrieval making the work highly relevant and transferable using transfer learning.

Acknowledgements We would like to thank the reviewers for their insightful comments which helped improve the overall quality of our manuscript.

Conflict of interest: The authors declare that they have no conflicts of interest.

References


1. Ayana, S.S., Liu, Z., Sun, M.: Neural headline generation with minimum risk training. arXiv preprint arXiv:1604.01904 (2016)
2. Bahdanau, D., Brakel, P., Xu, K., Goyal, A., Lowe, R., Pineau, J., Courville, A., Bengio, Y.: An actor-critic algorithm for sequence prediction. arXiv preprint arXiv:1607.07086 (2016)
3. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
4. Banko, M., Mittal, V.O., Witbrock, M.J.: Headline generation based on statistical translation. In: Proceedings of the 38th Annual Meeting on Association for Computational Linguistics, pp. 318–325. Association for Computational Linguistics (2000)
5. Carbonell, J.G., Goldstein, J.: The use of mmr, diversity-based reranking for reordering documents and producing summaries. In: SIGIR, vol. 98, pp. 335–336 (1998)
6. Chen, Y.C., Bansal, M.: Fast abstractive summarization with reinforce-selected sentence rewriting. arXiv preprint arXiv:1805.11080 (2018)
7. Cheng, J., Lapata, M.: Neural summarization by extracting sentences and words. arXiv preprint arXiv:1603.07252 (2016)
8. Cheung, J.C.K., Penn, G.: Unsupervised sentence enhancement for automatic summarization. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 775–786 (2014)
9. Chopra, S., Auli, M., Rush, A.M.: Abstractive sentence summarization with attentive recurrent neural networks. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 93–98 (2016)
10. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 (2014)

11. Clarke, J., Lapata, M.: Discourse constraints for document compression. *Computational Linguistics* **36**(3), 411–441 (2010)
12. Cohn, T., Lapata, M.: Sentence compression beyond word deletion. In: *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pp. 137–144. Association for Computational Linguistics (2008)
13. Colmenares, C.A., Litvak, M., Mantrach, A., Silvestri, F.: Heads: Headline generation as sequence prediction using an abstract feature-rich space. In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 133–142 (2015)
14. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018)
15. Dorr, B., Zajic, D., Schwartz, R.: Hedge trimmer: A parse-and-trim approach to headline generation. In: *Proceedings of the HLT-NAACL 03 on Text summarization workshop-Volume 5*, pp. 1–8. Association for Computational Linguistics (2003)
16. Erkan, G., Radev, D.R.: Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research* **22**, 457–479 (2004)
17. Fan, A., Grangier, D., Auli, M.: Controllable abstractive summarization. *arXiv preprint arXiv:1711.05217* (2017)
18. Filippova, K., Alfonseca, E., Colmenares, C.A., Kaiser, L., Vinyals, O.: Sentence compression by deletion with lstms. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 360–368 (2015)
19. Ganesan, K., Zhai, C., Han, J.: Opinosis: A graph based approach to abstractive summarization of highly redundant opinions. In: *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pp. 340–348 (2010)
20. Gehrmann, S., Deng, Y., Rush, A.M.: Bottom-up abstractive summarization. *arXiv preprint arXiv:1808.10792* (2018)
21. Grusky, M., Naaman, M., Artzi, Y.: Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. *arXiv preprint arXiv:1804.11283* (2018)
22. Henß, S., Mieskes, M., Gurevych, I.: A reinforcement learning approach for adaptive single-and multi-document summarization. In: *GSCL*, pp. 3–12 (2015)
23. Hsu, W.T., Lin, C.K., Lee, M.Y., Min, K., Tang, J., Sun, M.: A unified model for extractive and abstractive summarization using inconsistency loss. *arXiv preprint arXiv:1805.06266* (2018)
24. Jing, H., McKeown, K.R.: Cut and paste based text summarization. In: *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, pp. 178–185. Association for Computational Linguistics (2000)
25. Keneshloo, Y., Shi, T., Ramakrishnan, N., Reddy, C.K.: Deep reinforcement learning for sequence to sequence models. *arXiv preprint arXiv:1805.09461* (2018)
26. Kim, Y.: Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* (2014)
27. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014)
28. Knight, K., Marcu, D.: Statistics-based summarization-step one: Sentence compression. *AAAI/IAAI* **2000**, 703–710 (2000)
29. Lafferty, J., McCallum, A., Pereira, F.C.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data (2001)
30. Li, P., Bing, L., Lam, W.: Actor-critic based training framework for abstractive summarization. *arXiv preprint arXiv:1803.11070* (2018)
31. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: *Text summarization branches out*, pp. 74–81 (2004)
32. Lin, C.Y., Hovy, E.: Identifying topics by position. In: *Fifth Conference on Applied Natural Language Processing*, pp. 283–290 (1997)
33. Liu, Y.: Fine-tune bert for extractive summarization. *arXiv preprint arXiv:1903.10318* (2019)
34. Manly, B., McDonald, L., Thomas, D.L., McDonald, T.L., Erickson, W.P.: *Resource selection by animals: statistical design and analysis for field studies*. Springer Science & Business Media (2007)
35. Miao, Y., Yu, L., Blunsom, P.: Neural variational inference for text processing. In: *International conference on machine learning*, pp. 1727–1736 (2016)

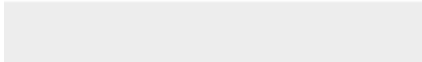

- 1 36. Microsoft: Attention spans (2015). URL <http://dl.motamem.org/microsoft-attention-spans-research-report.pdf>
- 2
- 3 37. Mohri, M., Pereira, F., Riley, M.: Weighted finite-state transducers in speech recognition. *Computer Speech & Language* **16**(1), 69–88 (2002)
- 4
- 5 38. Nallapati, R., Zhai, F., Zhou, B.: Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In: *Thirty-First AAAI Conference on Artificial Intelligence* (2017)
- 6
- 7 39. Nallapati, R., Zhou, B., Gulcehre, C., Xiang, B., et al.: Abstractive text summarization using sequence-to-sequence rnns and beyond. arXiv preprint arXiv:1602.06023 (2016)
- 8
- 9 40. Paulus, R., Xiong, C., Socher, R.: A deep reinforced model for abstractive summarization. arXiv preprint arXiv:1705.04304 (2017)
- 10
- 11 41. Mikolov, Tomas and Sutskever, Ilya and Chen, Kai and Corrado, Greg S and Dean, Jeff: Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, pp. 3111–3119 (2013)
- 12
- 13 42. Radev, D.R., Hovy, E., McKeown, K.: Introduction to the special issue on summarization. *Computational linguistics* **28**(4), 399–408 (2002)
- 14
- 15 43. Ranzato, M., Chopra, S., Auli, M., Zaremba, W.: Sequence level training with recurrent neural networks. In: *Iclr* (2016)
- 16
- 17 44. Research, G.V.: U.s. newspaper market size worth 17.07 billion by 2025 (2018). URL <https://www.grandviewresearch.com/press-release/us-newspaper-market-analysis>
- 18
- 19 45. Rush, A.M., Chopra, S., Weston, J.: A neural attention model for abstractive sentence summarization. arXiv preprint arXiv:1509.00685 (2015)
- 20
- 21 46. Ryang, S., Abekawa, T.: Framework of automatic text summarization using reinforcement learning. In: *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 256–265. Association for Computational Linguistics (2012)
- 22
- 23 47. Safran, N.: *Headline stats* (2015). URL <https://moz.com/blog/5-data-insights-into-the-headlines-readers-click>
- 24
- 25 48. Schulman, J., Moritz, P., Levine, S., Jordan, M., Abbeel, P.: High-dimensional continuous control using generalized advantage estimation. arXiv preprint arXiv:1506.02438 (2015)
- 26
- 27 49. See, A., Liu, P.J., Manning, C.D.: Get to the point: Summarization with pointer-generator networks. arXiv preprint arXiv:1704.04368 (2017)
- 28
- 29 50. Socher, R., Pennington, J., Huang, E.H., Ng, A.Y., Manning, C.D.: Semi-supervised recursive autoencoders for predicting sentiment distributions. In: *Proceedings of the conference on empirical methods in natural language processing*, pp. 151–161. Association for Computational Linguistics (2011)
- 30
- 31
- 32 51. Sutton, R.S., Barto, A.G.: *Reinforcement learning: An introduction* (2011)
- 33
- 34 52. Takase, S., Suzuki, J., Okazaki, N., Hiraio, T., Nagata, M.: Neural headline generation on abstract meaning representation. In: *Proceedings of the 2016 conference on empirical methods in natural language processing*, pp. 1054–1059 (2016)
- 35
- 36 53. Tan, J., Wan, X., Xiao, J.: From neural sentence summarization to headline generation: A coarse-to-fine approach. In: *IJCAI*, pp. 4109–4115 (2017)
- 37
- 38 54. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: *Advances in neural information processing systems*, pp. 5998–6008 (2017)
- 39
- 40 55. Vinyals, O., Fortunato, M., Jaitly, N.: Pointer networks. In: *Advances in Neural Information Processing Systems*, pp. 2692–2700 (2015)
- 41
- 42 56. Williams, R.J.: Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* **8**(3-4), 229–256 (1992)
- 43
- 44 57. Woodsend, K., Feng, Y., Lapata, M.: Generation with quasi-synchronous grammar. In: *Proceedings of the 2010 conference on empirical methods in natural language processing*, pp. 513–523. Association for Computational Linguistics (2010)
- 45
- 46 58. Yin, X.C., Pei, W.Y., Zhang, J., Hao, H.W.: Multi-orientation scene text detection with adaptive clustering. *IEEE transactions on pattern analysis and machine intelligence* **37**(9), 1930–1937 (2015)
- 47
- 48 59. Zajic, D., Dorr, B., Schwartz, R.: Automatic headline generation for newspaper stories. In: *Workshop on Automatic Summarization*, pp. 78–85 (2002)
- 49
- 50
- 51
- 52
- 53
- 54
- 55
- 56
- 57
- 58
- 59
- 60
- 61
- 62
- 63
- 64
- 65

-
- 1 60. Zeiler, M.D.: Adadelta: an adaptive learning rate method. arXiv preprint
2 arXiv:1212.5701 (2012)
 - 3 61. Zhou, L., Hovy, E.: Headline summarization at isi. In: Proceedings of the HLT-NAACL
4 2003 text summarization workshop and document understanding conference (DUC
5 2003), pp. 174–178 (2003)

6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65



Click here to access/download
Supplementary Material
spmpsi.bst






Click here to access/download
Supplementary Material
references.bib





Click here to access/download
Supplementary Material
SHEG Cover letter .doc





Click here to access/download
Supplementary Material
svglov3.clo





Click here to access/download
Supplementary Material
svjour3.cls





[Click here to access/download](#)

Supplementary Material

[Response_to_Reviewers_5thJun2020.pdf](#)

