Knowledge-Based Systems 240 (2022) 108108

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

STV-BEATS: Skip Thought Vector and Bi-Encoder based Automatic Text Summarizer

Minakshi Tomer [a,b,*], Manoj Kumar [c]

[a] USICT, GGSIPU, India
[b] IT Department, MSIT, Delhi, India
[c] Netaji Subhas University of Technology, East Campus (Formerly Ambedkar Institute of Advanced Communication Technologies and Research) Delhi, India

ARTICLE INFO

Article history:
Received 13 July 2021
Received in revised form 24 December 2021
Accepted 30 December 2021
Available online 8 January 2022

Keywords:
Abstractive text summarization
auto encoder
extractive text summarization
primary encoder
secondary encoder
skip thought vector

ABSTRACT

A novel text summarization framework referred to as Skip-Though Vector and Bi-encoder Based Automatic Text Summarization (STV–BEATS) is proposed in this paper. STV–BEATS utilizes — (a) skip-though vector to generate sentence-based embedding; and (b) Long Short-Term Memory (LSTM) based deep autoencoder to reduce dimensions of skip thought vectors. STV–BEATS works in the conjunction of extractive and abstractive summarization models to enhance the overall quality of the results. For each sentence, relevance and novelty metrics are calculated on the intermediate representation of the deep autoencoder to generate the final sentence score. The highly scored sentences are selected to generate an extractive summary. On the other hand, the abstractive part is composed of two encoders and a decoder which works as — (a) the first GRU-based bi-directional encoder and decoder work as basic sequence-to-sequence model on the extractive summary; and (b) the second GRU-based unidirectional encoder is used for fine encoding. Extensive computer experiments are conducted to determine the effectiveness of the STV–BEATS. Three standard benchmark datasets, namely, CNN/Daily Mail, DUC-2004, and DUC-2002 are used during experiments. Further, recall-oriented understudy for gisting evaluation (ROUGE) is used for validation of the STV–BEATS. Result reveals that the proposed STV–BEATS is capable of effective text summarization and achieves substantially better results over the state-of-the-art models.

1. Introduction

With the tremendous volume of textual data available on the internet which is constantly increasing at an exponential rate, it becomes a very challenging task to identify the key information present in the huge data. This results in a scenario where the user fails to grasp the main content as it is very time-consuming and even the data present is also redundant. One solution to this problem is shortening of text while maintaining its main content, but manual summarization of this data is not possible [1]. One of the vital solutions to this problem is automatic text summarization (ATS) which is a branch of natural language processing (NLP).

ATS is a process of producing a condensed text which pertains to all relevant information of large documents [2]. It helps the user to read and understand the content in a quick manner as it takes less space with less redundant information. Text summarization techniques on the basis of output are categorized as extractive and abstractive summarization [3]. In extractive methods keywords, key sentences, or objects are extracted without any modifications. This is an ad-hoc approach. It generally comprises three steps: intermediate representation of input, calculating sentence score, and sentence selection [4]. While abstractive method involves rephrasing of words, sentences, and generation of some new words to produce the final summary using different techniques [5]. Text summarization is also categorized as generic or query-oriented depending upon what is the aim of the user behind utilizing the summary. On the basis of input, it is categorized as single or multi-document summarization. It defines if the summary has to be produced from a single document or multiple documents. The other category is monolingual and multilingual summaries. In monolingual, the input and output are in the same language whereas, in multilingual input is in multilanguage and output is in one language.

In the past, abundant of research has been performed on extractive text summarization as it was easier to implement. A variety of traditional extractive summarization approaches are based on word frequency [6], cue word based [7], sentence length and position [8] to calculate the score of the sentences. Other extractive approaches include graph-based ranking [8,9], greedy approaches [10], LexRank [11], machine learning based and deep

* Corresponding author at: IT Department, MSIT, Delhi, India.
E-mail address: tomer.minakshi@gmail.com (M. Tomer).

https://doi.org/10.1016/j.knosys.2021.108108
0950-7051/

learning based [12] summarizer. A few researchers [5,13] implicate that extractive summarization has reached to its peak. Moreover, extractive summary lacks in terms of cohesion, readability, and even the difference between the human-generated summary and system summary is high [14]. Therefore, abstractive summarization and combination of extractive with abstractive has gained the attraction of researchers. Abstractive summarization can be classified as structure-based, semantic-based, discourse-based and deep neural network-based [5]. The methods based on deep neural networks outperform many state-of-art approaches in natural language processing tasks [15]. The abstractive summary is more human-readable, coherent, and rich with information content. Abstractive summary generates better results as compared to extractive, but it is complex and takes a lot of time to train the model. The previous sequence-to-sequence summarization system mainly depends on input data to generate summaries, which tends to work unstably. To overcome this limitation a hybrid model using an IR platform was utilized to retrieve the candidate template and extended sequence-to-sequence framework to generate the final summary [16].

In this paper, a hybrid framework is proposed which takes the benefit of both extractive and abstractive techniques. It is mainly based on SummCoder [12] for extractor and DEATS [17] for abstractor. In SummCoder skip-thoughts model was trained on an unlabelled Book Corpus dataset and a feed-forward neural network-based auto-encoder which was trained using fixed-length textual unit embeddings. For the extractive phase of the proposed framework skip thought vector [18] is used for sentence embedding which was trained on CNN/Daily mail dataset [19]. This vector representation of data is passed through an LSTM based autoencoder for generating better semantic relationships among the sentences. An LSTM based autoencoder handles sequential data of variable length and long-term dependencies. The output of this phase is fed as an input to the next phase which consists of two encoders and single decoder. The encoders used are Bi-directional GRU based RNN whereas the decoder is unidirectional GRU based RNN. The dataset used for training and testing the hybrid framework includes CNN/Daily mail [19], DUC-2004, and DUC-2002. The evaluation technique used is ROUGE [20].

The remaining paper is organized as follows: The literature survey of various extractive and abstractive techniques in the area of text summarization is mentioned in Section 2. Section 3 represented the research objective. Section 4 introduced the methodology for the proposed hybrid framework. The dataset used and the experimentation part is represented in Section 5. Section 6 concludes the paper with future scopes.

2. Literature survey

This section explains in detail survey of extractive methods and abstractive methods applied for text summarization.

2.1. Extractive text summarization

The first attempt towards text summarization was performed in 1958 [6] on a single document. In this the statistical information generated from the word frequency were used to identify the important sentences from the technical article. Later, in 1969 [7] concluded that apart from word frequency other factors should also be included like cue word, phrases, title words and sentence location.

Machine learning methods were successfully used for summary generation. It was started in 1995 [21] using binary classifier. Bayesian classifier methods were also used by [22,23] to select sentences. With the availability of annotated corpora, a wide range machine learning techniques were successfully applied in text summarization. Other methods include Hidden Markov [24], Decision Tree [23,25,26] and Integer linear programming [27].

An extractive summarization method to overcome redundancy by combining topic modelling and semantic measure was proposed by [28]. In this paper, two different methods i.e., combined topic vector and individual topic vector was also introduced. A graph extractive text summarization framework based on four algorithm was proposed in [29]. The first algorithm constructs a graph for input text whereas second and third algorithm selects the candidate sentences from the graph generated from first algorithm. The last algorithm limits the summary by selecting highly relevant sentences. Another graph-based summarization framework for single and multi-document was proposed by [30]. This framework utilizes semantic role labelling and explicit semantic analysis. The DUC-2002 dataset was used for experimentation part and Rouge metric for evaluation purpose. An extractive multi-document summarizer based on a combination of statistical, machine learning and graph-based method was proposed in [31]. In this clustering-based technique was used to identify the most important topic of the document on DUC-2002 and DUC-2006. A novel framework (SummCoder) for extractive text summarization of single document was proposed by [12]. The sentence level embedding and deep auto-encoders were utilized. Three metrics: sentence content relevance, sentence novelty and sentence position were used for sentence ranking and selection. A new dataset for text summarization named Illegal Document Summarization (TIDSumm) was also introduced.

Apart from these traditional methods, fuzzy logic techniques [32,33], swarm intelligence techniques [34] and deep learning techniques [35] were also successfully applied for extractive text summarization [36].

2.2. Abstractive text summarization

Abstractive approach for text summarization is majorly categorised as: (1) Structure-based (2) Semantic based (3) Discourse based and (4) Deep learning based [5].

The introduction of deep learning models brings a success to the area of NLP like machine translation, NER and dialogue system [37]. In a sentence level summarization, encoder with attention mechanism and decoder based on beam search is introduced by [38]. A simple encoder–decoder model which was proposed by [39] for machine translation is used by [40]. The model introduced a novel convolutional recurrent neural network with attention as encoder and RNN based decoder for abstractive sentence summarization. Both the encoder and decoder were trained on sentence summary pair dataset. This model outperforms the results on Giga word corpus and comparable results on DUC-2004 dataset. A novel architecture that extends the sequence-to-sequence model for abstractive summarization is proposed by [41]. This overcomes the limitation of earlier models via pointing and coverage mechanism. An encoder–decoder based model (deep recurrent generative decoder) proposed by [42] includes generative latent variables and discriminative deterministic states for abstractive text summary. In another model to improve the readability of generated summary an adversarial framework consists of a generator G and a discriminator D is proposed by [43]. The generator G takes the document as an input and use reinforcement learning for its optimization to generates the summary and discriminator D is a classifier that differentiate the system generated summary from the ground truth summary. In another paper [44] the word embedding based generator and attention-based discriminator was utilized to generate the text. For large documents and summary, a neural network model is proposed [45]. In this model the maximum likelihood training is replaced by reinforcement learning, encoder is bidirectional LSTM and decoder is unidirectional LSTM with an intra temporal attention. Another convolutional neural network summarization model is proposed by [46] which allows user to specify certain attributes like summary length, summary style, user interested entity in the generated summary. This model outperforms another summarization model. An extension of sequence-to-sequence framework for abstractive text summarization based on dual encoder (DEATS) was proposed by [17]. In this two encoders, primary and secondary encoders with a decoder was used. The primary encoder is bi-directional GRU to generate the feature representation of input. The secondary encoder is unidirectional GRU which generates importance weight and fine encoding based on input and previously generated output.

A two-phase approach for long text summarization (EA-LTS) was proposed by [47]. The extraction phase utilized a hybrid sentence similarity measure by sentence vector, Levenshtein distance, and integrate with graph model to select candidate sentences. For the abstraction phase, RNN based encoder–decoder with pointer and attention was used to generate the final summary. Another hybrid model for abstractive text summarization is proposed by [48] using CNN/Daily mail dataset, merged data of DUC 2003–2004 and DUC 2006–2007. This model utilizes fuzzy rules for extractive part and bi-directional LSTM for final summary generation. Another hybrid model of summarization and simplification using abstractor and extractor (HTSS) was proposed. In this a new metric CSS which combines the existing SARI and Rouge score was also proposed by [49].

3. Research objective

The main aim is to propose a framework that overcomes the limitations of extractive and abstractive summarization by combining these two approaches. The significant sentences are selected by the extractive phase thereby reducing redundancy and decreasing the size of the dataset for the abstractive phase. The abstractive phase re-phrases the sentences of the extractive summarization by generating new words, resulting in a summary more similar to a human-generated summary. The purpose of using extractive summary as input is to reduce the training time by providing more concise and quality input sentences.

In the extractive phase, sentence-based embedding is utilized instead of word embedding so that words in the more important sentences are more likely to be part of the extractive summary. This set of significant sentences are used by the LSTM based deep auto-encoder which manages the long-term dependencies. The main difference from the previous extractive model is that our extractive phase: is trained on a large benchmark CNN/Daily mail dataset which was together to provide more instances for better training. Secondly, instead of a feed-forward neural network, an LSTM based auto-encoder is utilized to handle the sequential data with variable length and finally, the summary generated by the extractor is an intermediate summary. For the abstractive phase instead of a single encoder and decoder, two encoders and a decoder are used. This abstractor takes the intermediate summary as input for better and quick context learning. The first encoder composed of bi-directional GRU computes the context vector using an attention mechanism with the decoder. The second encoder is a unidirectional GRU which calculates the importance weight of each input word and recalculates the context vector. The second encoder generates a more diverse summary based on how much importance is to be paid to an input word, its saliency to the entire context, and redundancy between the content source text and decoded content.

4. Proposed framework

In this section, a hybrid STV-BEATS (Skip thought vector Bi-encoder automatic text summarization) framework is proposed. It consists of three main phases. In the first phase, pre-processing is applied to clean the dataset. In the second phase, the extractive methodology is applied to generate an extractive summary using skip thought vector and deep autoencoder. In the third and last phase, the extractive summary is fed as an input to GRU-based bi-encoder and decoder for the final abstractive summary generation. This proposed framework can be represented by a block diagram as represented in Fig. 1.

4.1. Pre-processing

The pre-processing includes segmentation, tokenization, stop word removal, and stemming on the input data. The text is extracted from the documents, all the unnecessary data and tags are removed and then all the sentences are separated by a process called segmentation. All the words forming the sentences are extracted to individual tokens (tokenization). The cleaning of the document was done by removing the components (redundant white spaces, special characters, etc.) which have no contribution to the meaning of the input document called stop word removal. Sentences consisting of less than 5 words are omitted for the training of the network. The words are then reduced to their word stem by the process called stemming.

4.2. Extractive summarization phase

The main idea behind this phase is the generation of an intermediate extractive summary by applying a skip thought vector for sentence embedding instead of word embedding followed by an unsupervised technique called deep autoencoder for reducing the features. The final extractive summary generated is used as an input for the next phase.

4.2.1. Skip thought vectors

Skip-thought extends the skip-grams model from word embeddings to sentence embeddings. Instead of predicting context by surrounding words, skip-thought predicts the target sentence by using the previous sentence and the next sentence.

In the skip thought vector, two separate models were trained on the corpus. The first model consists of a unidirectional encoder with 2400 dimensions. The second is a bi-directional model consisting of two decoders: forward and backward, with each being of 1200 dimensions. The first decoder gets the input in the actual sequential order while for the other the sequence of input is reversed. These both are then used to get a resultant vector of 2400 dimensions by concatenation of both the outputs. Initialization of the matrices is done orthogonally for the training purpose. A uniform distribution of $[-0.1, 0.1]$ is used for the initialization of the non-recurrent weights. Optimization of the model is done with the help of the Adam optimizer. The gradients are clipped if the norms of the parameters are exceeded by 10 and the batch size used is 128. The combined model consists of the concatenated vectors of the previous two models resulting in a 4800-dimensional vector as shown in Fig. 2.

After the completion of training models, the expansion of the vocabulary was done. For this, the RNN encoder space was mapped with the word embeddings using CBOW word vectors. The vocabulary size which is used for the training of the skip thought vectors is set to be 20,000 words. The vocabulary of the CBOW model was further filtered by removing multiple word examples. This resulted in a vocabulary size of 930,911 words. With this expansion, the encoding is possible on 930,911 words even though the skip thought model was trained on relatively smaller vocabulary size.

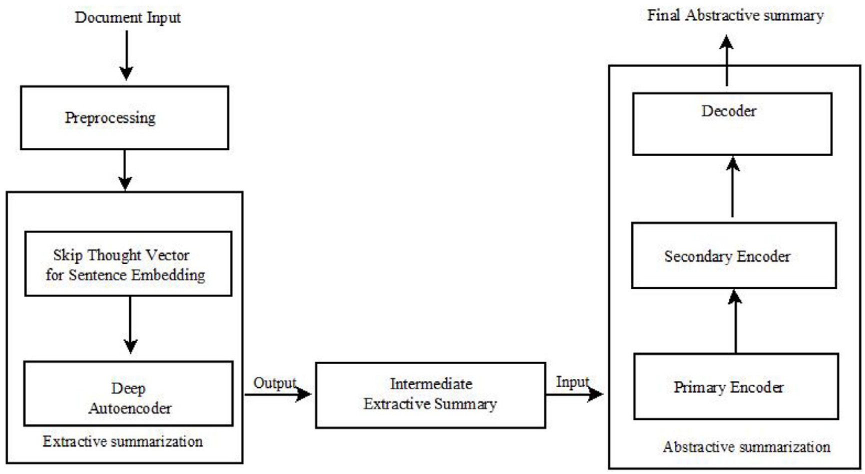

Fig. 1. The proposed STV-BEATS framework for automatic text summarization.

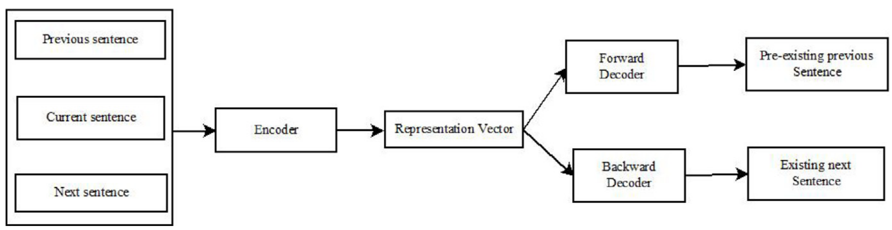

Fig. 2. Skip thought vector for extractive text summarization.

4.2.2. Deep autoencoders

An LSTM based autoencoder transforms the higher dimensional data into lower-dimensional representation via LSTM encoder at the hidden/bottleneck layer, and the LSTM decoder regenerates the higher dimensional representation. The functionality of the encoder is calculated as in Eq. (1).

$$z = f(\theta, x) = r(Wx + s) \tag{1}$$

where, the input feature vector is represented by x, the bottleneck feature is represented by z. θ =(W, s), W are the weights of the network and s is the bias. The activation function, either linear or non-linear, is represented by r.

The decoder maps the vector z of the bottleneck layer back to the higher dimensional representation, represented by Eq. (2).

$$y = g(\theta', z) = r(W'z + s') \quad where \qquad \theta' = (W', s') \tag{2}$$

The optimization of the autoencoder parameters is done by back-propagation between network output y and input x.

The two criteria are used to determine the quality of sentences to be included in the summary. The criteria are: sentence relevance score and sentence novelty score. To calculate the sentence relevance score $SentRev(D, S_i)$ as in Eq. (3), an intermediate representation of the whole document is created in the hidden layers of the autoencoder.

$$SentRev(D, S_i) = 1 - \frac{D.D_i'}{\|D\| \, . \, \|D'\|} \tag{3}$$

The output of the bottleneck layer for the complete document is represented as D, whereas $D_{i'}$ is the bottleneck layer output obtained after removing sentence i from the document. The sentence relevance score is calculated for each sentence in the document. The resultant value of the score is between [0, 1]. The higher the value the more relevant is the sentence to the summary.

The second parameter to identify the sentences to be included in the extractive summary is the sentence novelty score. The value generated by this parameter will be high if the sentence is repetitive or redundant and low otherwise. For the calculation of the novelty score first, the similarity between two sentences S_i and S_j of the document is computed using cosine similarity, then to compute the overall novelty of the sentence formula (4) is used.

$$SentNov(D, S_i) = \begin{cases} 1, & if\ max[Sim(S_i, S_j)] < \delta, \\ & 1 \le j \le N, i \ne j \\ 1, & if\ max[Sim(S_i, S_j)] > \delta, \\ & SentRev(D, S_i) > SentRev(D, S_k), \\ & k = argmax[Sim(S_i, S_j)], \\ & 1 \le j \le N, i \ne j \\ 1 - max[Sim(S_i, S_j)], & otherwise \end{cases} \tag{4}$$

The sentence novelty score $SentNov(D, S_i)$ is in range [0,1] and δ is the threshold value set to be 1. The value of score for sentence selection ranges from 0 to 1. The $max[Sim(S_i, S_j)]$ is the maximum similarity between any two sentences. The final score for the sentences are calculated by summing the sentence relevance score and sentence novelty score by using the formula (5) :

$$FinalScore(D, S_i) = \alpha.SentRev(D, S_i) + \beta.SentNov(D, S_i) \tag{5}$$

Here the fusion weights are represented by α, $\beta \in$ [0, 1] with $\alpha + \beta = 1$ [12]. They are used in the calculation of the final sentence score by assigning relative weights to different parameters. As the extractive summary is intermediate all relevant sentences are required for the better training of abstractive phase. The dimensionality of the intermediate summary is highly reduced. The main role of the intermediate summary is to provide context for the final summary. As the dimensionality is reduced the context is generated efficiently regardless of the position of

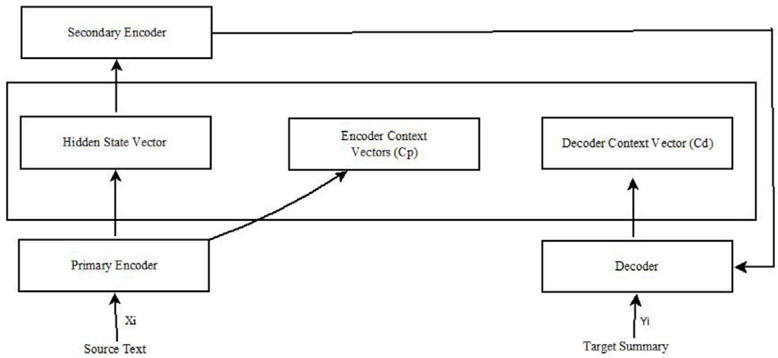

Fig. 3. Bi-encoder for abstractive text summarization.

the sentences. The average length of the intermediate summary is approximately one third (1/3rd) of the words of the source documents. An example of intermediate summary (Extractive summary) on CNN/daily mail dataset, DUC-2004 dataset, and DUC-2002 dataset is shown in Figs. 7, 8, and 9. The final step is the selection of sentences of higher ranks which will be used in the generation of the summary of a specific length.

4.3. Abstractive summarization phase

For the abstractive phase instead of a single encoder and decoder, two encoders and a decoder are used. This abstractor takes the intermediate summary as input for better and quick context learning. The first encoder composed of bi-directional GRU computes the context vector using an attention mechanism with the decoder. The second encoder is a unidirectional GRU which calculates the importance weight of each input word and recalculates the context vector. The second encoder generates a more diverse summary based on how much importance is to be given to an input word, its saliency to the entire context, and redundancy between the content source text and decoded content as presented in Fig. 3.

4.3.1. Primary encoder

The primary encoder is a bi-directional GRU which generates the features of the input sequence. GRU is the neural network equipped with the ability to capture the dependencies of different time scales which are mentioned in Eq. (6).

$$\begin{aligned} U_t &= \sigma(w_a[x_t, h_{t-1}]) \\ R_t &= \sigma(w_b[x_t, h_{t-1}]) \\ h'_t &= tanh(w_c[x_t, r_t \odot h_{t-1}]) \\ h_t &= (1-u_t) \odot h_{t-1} + u_t \odot h'_t \end{aligned} \tag{6}$$

The parameter matrices are represented by w_a, w_b and w_c, the hidden state vector and the input embedding vector at the given time step t is represented by h_t and x_t respectively and the multiplication operator applying the multiplication element wise is represented by $\odot$.

A sequence of input word embeddings $(x_1, x_2, x_j, \ldots, x_m)$ is fed to it. This is then used for the computation of hidden state representations $(\overrightarrow{h_1^p}, \overrightarrow{h_2^p}, \overrightarrow{h_j^p}, \ldots\ldots.\overrightarrow{h_m^p})$. The representations are computed at each word position sequentially in accordance with the current word embeddings. The previous hidden state representations for each word in reversed sequence $(\overleftarrow{h_1^p}, \overleftarrow{h_2^p}, \overleftarrow{h_j^p}, \ldots\ldots.\overleftarrow{h_m^p})$ are also computed by the same. Definition of both the hidden states are given in Eq. (7).

$$\begin{aligned} \overrightarrow{h_t^p} &= GRU^p(x_t, \overrightarrow{h_{t-1}^p}) \\ \overleftarrow{h_t^p} &= GRU^p(x_t, \overleftarrow{h_{t-1}^p}) \end{aligned} \tag{7}$$

$\overrightarrow{h_1^p} = 0$ and $\overleftarrow{h_m^p} = 0$ are initially set to zero vector, these are the initial states of the bidirectional GRU (Bi-GRU). Primary encoder reads the input sequence, after which the representation of each word sequence can be replaced by the concatenated hidden states of both the forward and backward GRU and is denoted as $h_t^p = [\overrightarrow{h_t^p}, \overleftarrow{h_t^p}]$. Non-linear transformation of the average pooling of the concatenated hidden states of the bidirectional GRU is then used to represent the whole input text sequence. C^z is used to represent it and is computed in Eq. (8).

$$C^z = tanh(W_k \frac{1}{N} \sum_{t=1}^{N} h_t^p + b_k) \tag{8}$$

N is the representation of the input sequence and the parameters are represented by W_k and b_k.

4.3.2. Secondary encoder

In the previous section, a primary encoder was utilized to read the input sequence and compute the context with attention mechanism. The secondary encoder is composed of unidirectional GRU instead of bi-directional GRU as in primary encoder. It takes the input sequence and calculates the importance weight μ_t. Its computation is based on various factors such as the representation of the feature of each word input h_t^i in the input sequence, the content representation of the output sequence C^o and the input sequence C^i generated by the decoder at current stage as shown in Eq. (9) [17].

$$\begin{aligned} \mu_t = \sigma(W_2(tanh(W_1[h_t^i, C^i, C^o] + b_1))) + h_t^{iT} W_s C^i \\ + h_t^{iT} W_s C^o - C^{iT} W_r C^o + b_2 \end{aligned} \tag{9}$$

The learning parameters are W_1, W_2, W_s, W_r, b_1 and b_2. The amount of importance to be given to the current input word x_t is given by importance weight μ_t. $h_t^{iT} W_s C^i$ and $h_t^{iT} W_s C^o$ gives the attention to each word of the entire content of source text. $C^{iT} W_r C^o$ represents the redundancy between the decoded content in the current stage and the content of the original text. The importance weight is calculated based on the redundancy, the saliency and the context of the information as various factors.

The information flow is biased by putting the importance weight on skip connections. This is done to check whether the information on the current hidden state h_t^m must be considered by the secondary encoder or whether majority of the information will be taken from the previous hidden state h_{t-1}^m. This is done by comparing the value of importance weight with value of current input word. If the value is less, then the information is taken from the previous hidden state otherwise not. And if the value turns out to be unity i.e. 1 then the encoder will be influenced by the current state of the input because then it will act as a standard

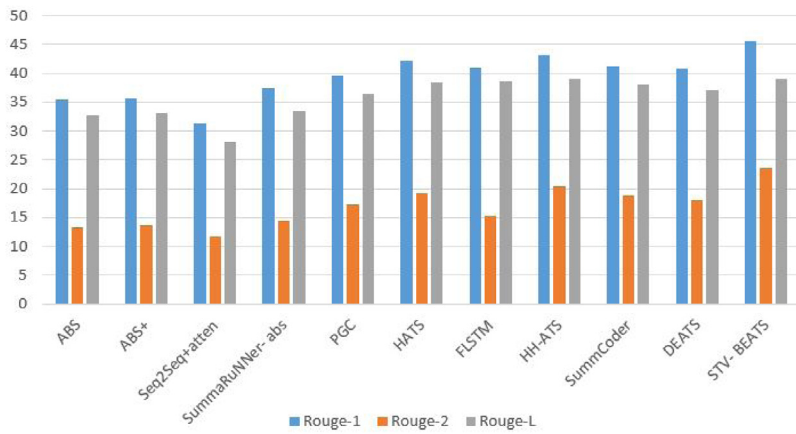

Fig. 4. Rouge score on CNN/Daily Mail dataset.

GRU cell. The following rule will be followed to update the value as in Eq. (10).

$$h_t^m = (1 - \mu_t) \odot h_{t-1}^m + \mu_t \odot GRU^s(x_t, h_{t-1}^m) \tag{10}$$

This hidden state information assist decoder to generate final summary. The dual encoding process is completed by passing the text through both the primary and the secondary encoder.

4.3.3. Decoder

The decoder is GRU based with attention mechanism. The primary encoder and the decoder performs the basic sequence-to-sequence approach. The role of secondary encoder is an additional and independent encoder used to improve the performance of the basic model. The context vector is computed by taking the hidden states of the primary encoder into consideration and attention mechanism is used with the decoder. The weighted sum of each hidden states marks the computation of the context vector C_v as in Eq. (11).

$$C_v = \sum_{j=1}^{n} a_{ij} h_j^p \tag{11}$$

Here, h_j^p is hidden state whose weight a_{ij} is computed as in Eq. (12).

$$\begin{aligned} a_{ij} &= \frac{exp(e_{ij})}{\sum_{k=1}^{n} exp(e_{ik})} \\ e_{ij} &= v_a^T tanh(W_a h_{i-1}^k + U_a h_j^p) \\ h_i^k &= GRU^d(y_i, h_{i-1}^k) \end{aligned} \tag{12}$$

Matching ratio of the inputs around position j to the output position i is represented by e_{ij}. h_i^k represents the hidden state generated by the decoder. The last hidden state h_{i-1}^k and the i_{th} target y_i in the output sequence are the two factors on which the hidden state generated by decoder will depend.

In this the decoding takes place in stages rather than the whole document being decoded in a one go. The decoding begins by the decoding of partial length sequence of fixed length t and then the whole sequence decoded is calculated as in Eq. (13).

$$C_r = tanh(W_r \frac{1}{L} \sum_{i=1}^{L} h_i^p + b_r) \tag{13}$$

The learnable parameters are W_r, b_r and the current length of decoded sequence is represented by L. C_r represents the current decoded content which is used for the adjustment of the attention weight of secondary encoder. A new final state is generated by the secondary encoder, after every fixed length decoding, which is represented by h_m^s. The decoder is now represented as in Eq. (14).

$$h_i^k = \begin{cases} GRU^d(y_i, [h_{i-1}^k, h_m^s]) & if\ L\%t == 0 \\ GRU^d(y_i, h_{i-1}^k) & otherwise \end{cases} \tag{14}$$

The final state of the primary encoder acts as the initial state of the decoder i.e., $h_0^k = h_m^p$. Computation of secondary encoding and decoded content takes place at every t decoding steps. Then the concatenation of the current vector C_v, acquired from the primary encoder, and the hidden state of the decoder h_i^k takes place. The vocabulary distribution is produced when the resultant is fed through a linear layer and is computed as in Eq. (15).

$$P(y_i|y_1, \ldots.., y_{i-1}; x) = softmax(W_z[h_i^k, C_v] + b_z) \tag{15}$$

Here, the conditional probability distribution is denoted by $P(y_i|y_1, \ldots.., y_{i-1}; x)$ and it is found for the target word y_i over all the words in the given vocabulary at the given time step i. The learnable parameters are denoted by W_z and b_z.

5. Experimental setup and results

This section begins with the introduction of the datasets used, the system setting for the experimentation purpose, and the evaluation metric used to evaluate the proposed framework. Then the experimental results of the STV-BEATS are compared with state-of-art models. The workstation used was equipped with a Ryzen 7 3700x (8 cores 16 threads) CPU clocked at 4.4 GHz, has 16 Gb of ddr4 memory clocked at 3600 MHz. The GPU is a rtx 2070 super with 8 Gb of gddr6 memory, has 2560 Cuda cores and 320 tensor cores. GPUs are optimized for training deep learning models as they can process multiple computations simultaneously.

5.1. Dataset and evaluation metric

There are multiple standard datasets available for summarization purposes. For the experiments, CNN/Daily Mail dataset [19] which is a benchmark data for abstractive text summarization is used. It contains the news article with multiple sentences as a reference summary. Both the datasets i.e., CNN and Daily Mail are combined to provide more data instances for better training. The combined dataset is divided into three sets, training, testing, and evaluation. For training 249,337 articles were used, for testing as well as validation 311,672 articles were used.

DUC-2002 and DUC-2004 are also used which is a part of the Document understanding conference and generated by NIST (National Institute of Standard and Technology). The URL of the

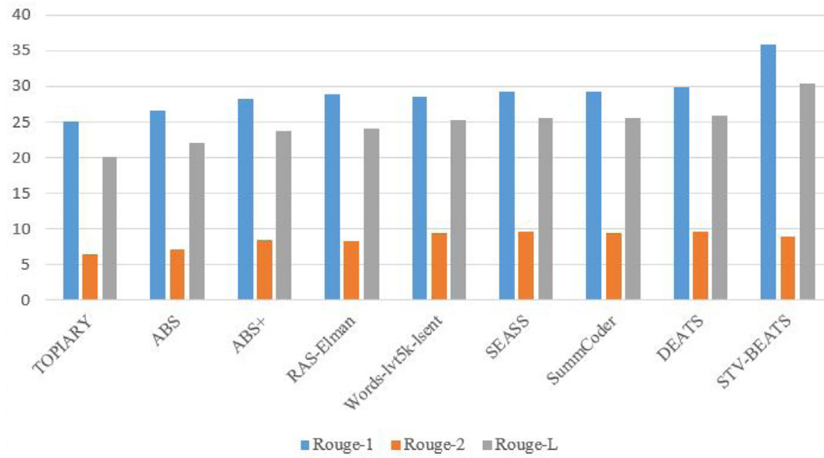

Fig. 5. Rouge score on DUC-2004 dataset.

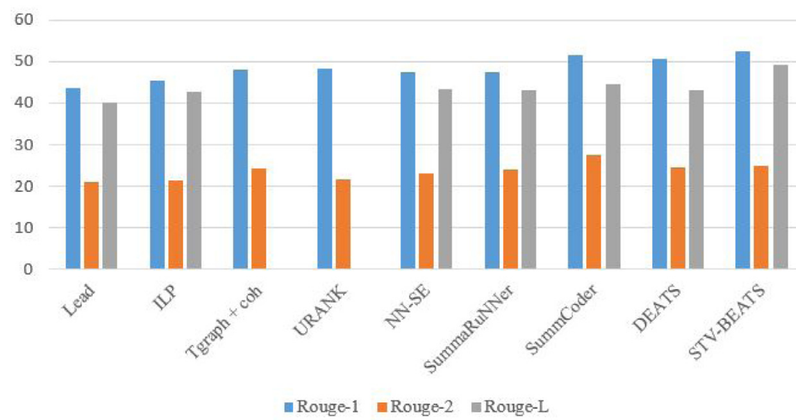

Fig. 6. Rouge score on DUC-2002 dataset.

Table 1
Characteristics of Dataset for Text summarization.

Dataset	Domain	Number of documents	Summary length
DUC 2002	News	567	100 words
DUC 2004	News	500	100 words
CNN dataset	News	90,000	–
Daily Mail dataset	News	197,000	–

dataset is http://www-nlpir.nist.gov/projects/duc/data. Both the corpus are used as a testing dataset to evaluate the summary. Characteristics of the datasets for text summarization are presented in Table 1. The evaluation is performed using ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [20] to measure the quality of summary by calculating the number of overlapping words between the gold standard summary and system-generated summary. This is the most commonly used metrics for text summarization which include ROUGE-1, ROUGE-2, and ROUGE-L.

5.2. Experimental setting

CNN/Daily Mail dataset is used for the training and the tuning of the model. The embeddings size is kept at 2400 because with this size the results produced by various authors were state-of-art. The retraining of the model is a necessary step to generate embeddings of different size. Then the generated embeddings must be verified on different NLP tasks before it is applied on summarization. The deep autoencoders are also trained on the previously mentioned dataset (CNN/Daily Mail). The evaluation and testing are carried on the DUC datasets.

The dimension of the hidden state is set to be 256 and the dimension of word embeddings is 128. The vocabulary size is set to 50k words. The learning rate is set to be at 0.1. The token size is kept as 120 at the time of training and testing. This is done to accelerate the training procedure and the output summaries are decoded with the help of beam search with the size of beam set to 4. Random initialization of the network parameters is done over a uniform distribution within [−0.05, 0.05]. Furthermore, Adagrad algorithm is used for optimization.

The ability to handle OOV words on its own makes the model versatile for the use of a smaller vocab size. Moreover, the word embeddings did not pertained rather they were learned from scratch during the training process.

5.3. Results and discussion

In this section proposed framework (STV-BEATS) is compared with other state-of-art models on different datasets which are described below.

(1) CNN/Daily Mail dataset: ABS and ABS+ [38] is a model which is implemented with standard encoder–decoder with an attention mechanism. Other models are seq2seq+atten [39] which is a standard sequence-to-sequence model with attention, SummaRuNNer-abs [35] is RNN based model which is converted

CNN/ Daily-Mail
Article
Diners who attended President Barack Obama 's evening meal Monday were treated to a three-course menu of fusion fare with hints of the subcontinent. Not partaking: the guest of honor, Indian Prime Minister Narendra Modi, who is midway through a strict religious fast. After encouraging other guests to eat, he sipped only warm water. The timing of Modi 's visit which coincides with the Hindu festival devoted to the goddess Shakti, was not expected to deter from the high-level discussions on trade and security, White House officials said before Modi 's arrival. And on Monday evening, guests including Vice President Joe Biden and Secretary of State John Kerry did enjoy a gourmet meal from the White House kitchen during the working dinner. Opinion: Why Modi 's visit matters. The menu included an avocado salad with goat cheese, crisped halibut with carrot ginger sauce and basmati rice, and mango crème brûlée. Ahead of Modi 's trip to the United States, which includes a whopping 50 stops to visit with CEOs and a speech at the United Nations, officials said he intended to survive solely on `` nimbu pani '' -- or water with lemon -- for nine days. At the White House Monday, Modi presented Obama with a copy of `` The Bhagavad Gita According to Gandhi, '' a Hindu religious text that was annotated and translated by the Indian independence leader Mahatma Gandhi. The Indian Ministry of External Affairs posted several photos from the White House proceedings
Standard Summary
President Barak Obama gave a dinner in the honour of Indian Prime Minister. Modi was midway through a fast in regards for the hindu festival devoted to goddess shakti. Modi encouraged others to eat while he only drank warm water. Vice President Joe Biden and Secretary of State John Kerry also joined the dinner. The visit was important as he was to visit 50 CEOs and a speech at United Nations. Moreover, he intended to survive solely on lemon water. The dinner consisted of many fruits such as Avocado, mangoes etc. Modi ji gifted The Bhagavad Gita According to Gandhi, translation of the holy book Gita according to Gandhi Ji. The Indian ministry of external affairs shared many pictures of the visit.
Extractive Summary
And on Monday evening, guests including Vice President Joe Biden and Secretary of State John Kerry did enjoy a gourmet meal from the White House kitchen during the working dinner...Ahead of Modi's trip to the United States, which includes a whopping 50 stops to visit with CEOs and a speech at the United Nations , officials said he intended to survive solely on nimbu pani '' -- or water with lemon -- for nine days..."
Abstractive Summary
guests including vice president Joe Biden and secretary of state John Kerry enjoyed a gourmet meal from the white house kitchen during the working dinner. Officials said he intended to survive solely on nimbu pani '' or water with lemon -- for nine days..." Modi's trip to the u.s. includes 50 stops to visit with CEOs and speech at the united Nations and an inauguration speech for president Obama on Monday

Fig. 7. Example of Abstractive Summary Generated from Proposed Framework (STV-BEATS) using Extractive summary as input On CNN/Daily Mail dataset.

from an extractive model, PGC [41] is a standard sequence-to-sequence model with coverage to overcome OOV word problem, HATS [50] is a hybrid learning model which simulates the technique of human method. It is also compared with FLSTM [48] which is an ensembled model implemented using the fuzzy and LSTM technique, HH-ATS [51] is a human-like deep network composed of knowledge module, multitask module and DD-GAN. SummCoder [12] an extractive summarization model and DEATS [17] which is a model for abstractive text summarization implemented by two encoders and a decoder.

(2) DUC-2004: TOPIARY [52] which is a topic detection algorithm, ABS and ABS+ [38] which is the original model for machine translation, RAS-Elman [40] in which the encoder uses attention and decoder is based on recurrent neural network, words-lvt5k-lsent [53] is another attention-based encoder–decoder model, SEASS [54] is a encoder–decoder model with selective gate network,SummCoder [12] an extractive text summarization method and DEATS [17].

(3) DUC-2002: The base model Lead which selects the key sentences, ILP [55] is a phrase-based summarization technique using integer linear programming. Other traditional models based on graph-based techniques Tgraph [56] which uses Latent Dirichlet Allocation and URANK [57] is unified rank methodology. It is also compared with NN-SE [58] neural-based summarization for sentence selection, SummaRuNNer [35] is a recurrent neural network model and SummCoder [12] which is a extractive summarization model with best state of art result among other models and DEATS.

The experimental results of STV-BEATS on the CNN/Daily Mail dataset is presented in Table 2 and Fig. 4. It achieves high ROUGE scores on CNN/Daily mail as compared to other models. The value of ROUGE-1 scores as 45.53, ROUGE-2 score as 23.52, and ROUGE-L score as 39.10 is attained. The results clearly indicate that the proposed method outperforms many other existing methods. The evaluation result on DUC-2004 achieves a high ROUGE-1 score as 35.91 and ROUGE-L score as 30.38 and comparable ROUGE-2 score as 8.96 as shown in Table 3 and Fig. 5. On the other hand, it also achieves a high ROUGE score on the DUC-2002 dataset. The ROUGE-1 score is 52.35, ROUGE-L score is 49.23 and comparable ROUGE-2 score as 24.8 as shown in Table 4 and Fig. 6.

The evaluation achieves improvement in all ROUGE score for CNN/Daily Mail dataset as this dataset is comparatively larger than DUC. This is the possible reason to use CNN/Daily Mail dataset for training the model. A qualitative analysis on the different datasets indicates its effectiveness. In Fig. 7 for an input data (CNN/Daily Mail dataset) of length 800–820 words (approximately) the length of standard summary is about 60–63 words. In Fig. 8 for an input data (DUC-2004) of length 400–410 words (approximately) the length of standard summary is about 100–110 words. In Fig. 9 for an input data (DUC-2002) of length 400–410 words (approximately) the length of standard summary is about 100–110 words.

The length of intermediate summary (Extractive summary) is almost one third (1/3rd) of input document whereas abstractive summary is reduced to one fourth (1/4th) of input document with generation of new words and sentence reformation.

6. Conclusion and future scope

In this paper, a hybrid extractive abstractive based framework (STV-BEATS) which enhanced the traditional extractive and abstractive model is proposed for automatic text summarization. It generates extractive summary using sentence level embedding and deep-auto encoder for better semantic representation of sentences. The extractive summary is fed as input to the two-level

DUC 2004 (Article)

Cambodia's two-party opposition asked the Asian Development Bank Monday to stop providing loans to the incumbent government, which it calls illegal. Negotiations to form the next government have become deadlocked, and opposition party leaders Prince Norodom Ranariddh and Sam Rainsy are out of the country following threats of arrest from strongman Hun Sen. Hun Sen complained Monday that the opposition was trying to make its members' return an international issue. Hun Sen's ruling party narrowly won a majority in elections in July, but the opposition claiming widespread intimidation and fraud has denied Hun Sen the two-thirds vote in parliament required to approve the next government. Meanwhile, it says, the old government is holding power illegally. Ranariddh and Sam Rainsy renewed their international lobbying campaign against the old government Monday in a letter to ADB President Mitsuo Sato calling for the bank to stop lending money to it. ``We respectfully advise the Asian Development Bank not to provide any new loans to the current regime in Cambodia," the two-party leaders wrote. ``At best the current regime could be considered a caretaker government as it has not been approved by the National Assembly." After a meeting between Hun Sen and the new French ambassador to Cambodia, Hun Sen aide Prak Sokhonn said the Cambodian leader had repeated calls for the opposition to return, but expressed concern that the international community may be asked for security guarantees. ``There have been reports that there is an attempt to internationalize the return of those members of parliament on the excuse of security problems," Prak Sokhonn said. ``Some (opposition politicians) have wanted the United Nations to help guarantee a safe return for them." The U.N. secretary-general's representative office in Phnom Penh provided monitors to opposition politicians after they returned to Cambodia to participate in the July election. The monitoring ended Sept. 30. ``Our office has not received any official request for that operation to be started up again," U.N. diplomat Jonathan Prentice said Monday in reaction to Prak Sokhonn's statement. The opposition has insisted that any further talks on the next government must take place outside the country, but the ruling party has rejected allegations of intimidation and recently guaranteed opposition members' safety inside the country. Diplomatic efforts to revive the stalled talks appeared to bear fruit Monday as Japanese Foreign Affairs Secretary of State Nobutaka Machimura said King Norodom Sihanouk has called on Ranariddh and Sam Rainsy to return to Cambodia. Less than two weeks after abandoning hope that he could influence the parties to reach a compromise, Sihanouk is now ``strongly interested" in presiding over a summit meeting of the three-party leaders in Cambodia, Machimura said.

Standard Summary

Cambodia King Norodom Sihanouk praised formation of a coalition of the Countries top two political parties, leaving strongman Hun Sen as Prime Minister and opposition leader Prince Norodom Ranariddh president of the National Assembly. The announcement comes after months of bitter argument following the failure of any party to attain the required quota to form a government. Opposition leader Sam Rainey was seeking assurances that he and his party members would not be arrested if they return to Cambodia. Rainey had been accused by Hun Sen of being behind an assassination attempt against him during massive street demonstrations in September.

Extractive Summary

Cambodia's two-party opposition asked the Asian Development Bank Monday to stop providing loans to the incumbent government, which it calls illegal. Hun Sen's ruling party narrowly won a majority in elections in July, but the opposition _ claiming widespread intimidation and fraud _ has denied Hun Sen the two-thirds vote in parliament required to approve the next government. Meanwhile, it says, the old government is holding power illegally. ``At best the current regime could be considered a caretaker government as it has not been approved by the National Assembly." Diplomatic efforts to revive the stalled talks appeared to bear fruit Monday as Japanese Foreign Affairs Secretary of State Nobutaka Machimura said King Norodom Sihanouk has called on Ranariddh and Sam Rainsy to return to Cambodia. Less than two weeks after abandoning hope that he could influence the parties to reach a compromise, Sihanouk is now ``strongly interested" in presiding over a summit meeting of the three-party leaders in Cambodia, Machimura said.

Abstractive Summary

two-party opposition asks the Asian Development Bank to stop lending to the incumbent government, which it calls illegal. the opposition says the old government is holding power illegally, claiming widespread intimidation and fraud. 'At best the current regime could be considered a caretaker government as it has not been approved by the National Assembly'

Fig. 8. Example of Abstractive Summary Generated from Proposed Framework (STV-BEATS) using Extractive summary as input On DUC-2004 dataset.

Table 2
Performance comparison of several model on CNN/Daily Mail dataset.

Method	Rouge-1	Rouge-2	Rouge-L
ABS	35.46	13.3	32.65
ABS+	35.63	13.75	33.01
Seq2Seq+atten	31.34	11.79	28.1
SummaRuNNer- abs	37.5	14.5	33.4
PGC	39.53	17.28	36.38
HATS	42.16	19.17	38.35
FLSTM	40.96	15.22	38.63
HH-ATS	43.16	20.32	39.1
SummCoder	41.2	18.78	37.96
DEATS	40.85	18.08	37.13
STV- BEATS	**45.53**	**23.52**	**39.1**

encoders and decoders for the final abstractive summary. This combination of the extractive and abstractive model provides benefit to each other in terms of efficiency. The experimental results on CNN/Daily Mail dataset represent that the method achieves better ROUGE scores than other state-of-art models. The results on DUC datasets are also better in terms of ROUGE-1 and ROUGE-L scores. Whereas, it generates a comparable ROUGE-2 score in comparison to other traditional models.

In the future, another extractive model can be trained with sentence embedding to improve the F1 score. This can be further integrated with other enhanced abstractive models like

Table 3
Performance comparison of several model on DUC-2004 dataset.

Method	Rouge-1	Rouge-2	Rouge-L
TOPIARY	25.12	6.46	20.12
ABS	26.55	7.06	22.05
ABS+	28.18	8.49	23.81
RAS-Elman	28.97	8.26	24.06
Words-lvt5k-lsent	28.61	9.42	25.24
SEASS	29.21	9.56	25.51
SummCoder	29.31	9.54	25.67
DEATS	29.91	9.61	25.95
STV-BEATS	**35.91**	8.96	**30.38**

Table 4
Performance comparison of several model on DUC-2002 dataset.

Method	Rouge-1	Rouge-2	Rouge-L
Lead	43.6	21	40.2
ILP	45.4	21.3	42.8
Tgraph + coh	48.1	24.3	–
URANK	48.5	21.5	–
NN-SE	47.4	23	43.5
SummaRuNNer	47.4	24	43.03
SummCoder	51.7	27.5	44.6
DEATS	50.8	24.6	43.2
STV-BEATS	**52.35**	**24.8**	**49.23**

DUC 2002 Article

in Spain, the national anarcho-syndicalist trade union confederación national del trabajo initially refused to join a popular front electoral alliance, and abstention by cnt supporters led to a right-wing election victory. but in 1936, the cnt changed its policy and anarchist votes helped bring the popular front back to power. months later, the former ruling class responded with an attempted coup causing the Spanish civil war -lrb- 1936 -- 1939 -rrb- . in response to the army rebellion, an anarchist-inspired movement of peasants and workers, supported by armed militias, took control of Barcelona and of large areas of rural spain where they collectivised the land. but even before the fascist victory in 1939, the anarchists were losing ground in a bitter struggle with the Stalinists, who controlled much of the distribution of military aid to the republican cause from the Soviet Union. the events known as the Spanish revolution was a worker ' social revolution that began during the outbreak of the Spanish civil war in 1936 and resulted in the widespread implementation of anarchist and more broadly libertarian socialist organisational principles throughout various portions of the country for two to three years, primarily Catalonia, Aragon, Andalusia, and parts of the levante. much of spain's economy was put under worker control; in anarchist strongholds like catalonia, the figure was as high as 75 %, but lower in areas with heavy communist party of spain influence, as the soviet-allied party actively resisted attempts at collectivization enactment. factories were run through worker committees; agrarian areas became collectivised and run as libertarian communes. anarchist historian sam dolgoff estimated that about eight million people participated directly or at least indirectly in the Spanish revolution, which he claimed `` came closer to realizing the ideal of the free stateless society on a vast scale than any other revolution in history. '' Stalinist-led troops suppressed the collectives and persecuted both dissident marxists and anarchists. the prominent Italian anarchist Camillo berneri, who volunteered to fight against franco was killed instead in spain by gunmen associated with the Spanish communist party.
post-war years.

Standard Summary

The national trade union refused to join a popular front electoral alliance. This led to their victory. In 1936 the rules were changed by cnt and with the help of anarchist votes they came back in power. The union took part in a coup which caused the Spanish civil war in 1936. This moment of peasants and workers was supported by armed militants which then took control of major areas of rural spain. This was not easy as it was seen. The widespread of the movement was halted in its initial stages as the communist party had a say in over 75% of the area. The economy was struck. As the factories were run by worker union, they came closer to the truth and almost 8 million people took part in the movement either directly and indirectly. This sparked the light and the revolution spread. Stalin led the troops to stop this revolt and was very successful in doing s. the Italian anarchist Camillo berneri was killed in spain by the gunmen of the communist party

Extractive Summary

in spain, the national anarcho-syndicalist trade union confederación nacional del trabajo initially refused to join a popular front electoral alliance, and abstention by cnt supporters led to a right-wing election victory. months later, the former ruling class responded with an attempted coup causing the Spanish civil war -lrb- 1936 -- 1939 -rrb-. much of spain 's economy was put under worker control; in anarchist strongholds like catalonia, the figure was as high as 75 %, but lower in areas with heavy communist party of spain influence, as the soviet-allied party actively resisted attempts at collectivization enactment. the prominent Italian anarchist Camillo berneri, who volunteered to fight against franco was killed instead in spain by gunmen associated with the Spanish communist party.

Abstractive Summary

in spain, the national anarcho-syndicalist trade union confederación national del trabajo initially refused to join a popular front electoral alliance. the former ruling class responded with an attempted coup causing the Spanish civil war -lrb- 1936 -- 1939 -- rrf-, which was thrown under worker control in the u.s the figure was high in strongholds but lower in areas with heavy influence of communist party of spain. Camillo berneri who volunteered to fight against franco was killed by gunmen associated with Spanish communist party.

Fig. 9. Example of Abstractive Summary Generated from Proposed Framework (STV-BEATS) using Extractive summary as input On DUC-2002 dataset.

reinforcement-based learning, transformers and hybrid training functions to improve the overall performance of the hybrid model.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] G.C.V. Vilca, M.A.S. Cabezudo, A study of abstractive summarization using semantic representations and discourse level information, in: International Conference On Text, Speech, And Dialogue, Springer, 2017, pp. 482–490.
[2] E. Lloret, Text summarization: an overview, Paper Supported By The Spanish Government Under The Project TEXT-MESS (TIN2006-15265-C06-01), 2008.
[3] E. Lloret, M. Palomar, Text summarisation in progress: a literature review, Artif. Intell. Rev. 37 (1) (2012) 1–41.
[4] A. Nenkova, K. McKeown, A survey of text summarization techniques, in: Mining Text Data, Springer, 2012, pp. 43–76.
[5] S. Gupta, S. Gupta, Abstractive summarization: An overview of the state of the art, Expert Syst. Appl. 121 (2019) 49–65.
[6] H.P. Luhn, The automatic creation of literature abstracts, IBM J. Res. Dev. 2 (2) (1958) 159–165.
[7] H.P. Edmundson, New methods in automatic extracting, J. ACM (JACM) 16 (2) (1969) 264–285.
[8] G. Erkan, D.R. Radev, Lexrank: Graph-based lexical centrality as salience in text summarization, J. Artif. Intell. Res. 22 (2004) 457–479.
[9] R.C. Belwal, S. Rai, A. Gupta, A new graph-based extractive text summarization using keywords or topic modeling, J. Ambient Intell. Humaniz. Comput. (2020) 1–16.
[10] J. Carbonell, J. Goldstein, The use of MMR, diversity-based reranking for reordering documents and producing summaries, in: Proceedings Of The 21st Annual International ACM SIGIR Conference On Research And Development In Information Retrieval, 1998, pp. 335–336.
[11] R. Barzilay, M. Elhadad, Using lexical chains for text summarization, in: Advances In Automatic Text Summarization, 1999, pp. 111–121.
[12] A. Joshi, E. Fidalgo, E. Alegre, L. Fernández-Robles, SummCoder: An unsupervised framework for extractive text summarization based on deep auto-encoders, Expert Syst. Appl. 129 (2019) 200–215.
[13] P. Mehta, From extractive to abstractive summarization: A journey, in: ACL (Student Research Workshop), Springer, 2016, pp. 100–106.
[14] J.-g. Yao, X. Wan, J. Xiao, Recent advances in document summarization, Knowl. Inf. Syst. 53 (2) (2017) 297–336.
[15] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, P. Kuksa, Natural language processing (almost) from scratch, J. Mach. Learning Res. 12 (ARTICLE) (2011) 2493–2537.
[16] Z. Cao, W. Li, F. Wei, S. Li, et al., Retrieve, rerank and rewrite: Soft template based neural summarization, 2018, Association for Computational Linguistics (ACL).
[17] K. Yao, L. Zhang, D. Du, T. Luo, L. Tao, Y. Wu, Dual encoding for abstractive text summarization, IEEE Trans. Cybern. 50 (3) (2018) 985–996.
[18] R. Kiros, Y. Zhu, R. Salakhutdinov, R.S. Zemel, A. Torralba, R. Urtasun, S. Fidler, Skip-thought vectors, 2015, arXiv preprint arXiv:1506.06726.
[19] K.M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, P. Blunsom, Teaching machines to read and comprehend, Adv. Neural Inf. Process. Syst. 28 (2015) 1693–1701.
[20] C.-Y. Lin, Rouge: A package for automatic evaluation of summaries, in: Workshop On Text Summarization Branches Out (WAS 2004), 2004.
[21] J. Kupiec, J. Pedersen, F. Chen, A trainable document summarizer, in: Proceedings Of The 18th Annual International ACM SIGIR Conference On Research And Development In Information Retrieval, 1995, pp. 68–73.

[22] S. Teufel, Sentence extraction as a classification task, in: Intelligent Scalable Text Summarization, 1997.
[23] J.L. Neto, A.A. Freitas, C.A. Kaestner, Automatic text summarization using a machine learning approach, in: Brazilian Symposium On Artificial Intelligence, Springer, 2002, pp. 205–215.
[24] J.M. Conroy, D.P. O'leary, Text summarization via hidden markov models, in: Proceedings Of The 24th Annual International ACM SIGIR Conference On Research And Development In Information Retrieval, 2001, pp. 406–407.
[25] I. Mani, E. Bloedorn, Machine learning of generic and user-focused summarization, in: AAAI/IAAI, 1998, pp. 821–826.
[26] K. Knight, D. Marcu, Summarization beyond sentence extraction: A probabilistic approach to sentence compression, Artificial Intelligence 139 (1) (2002) 91–107.
[27] W. Luo, F. Liu, Z. Liu, D. Litman, A novel ILP framework for summarizing content with high lexical variety, 2018, arXiv preprint arXiv:1807.09671.
[28] R.C. Belwal, S. Rai, A. Gupta, Text summarization using topic-based vector space model and semantic measure, Inf. Process. Manage. 58 (3) (2021) 102536.
[29] W.S. El-Kassas, C.R. Salama, A.A. Rafea, H.K. Mohamed, EdgeSumm: Graph-based framework for automatic text summarization, Inf. Process. Manage. 57 (6) (2020) 102264.
[30] M. Mohamed, M. Oussalah, SRL-ESA-TextSum: A text summarization approach based on semantic role labeling and explicit semantic analysis, Inf. Process. Manage. 56 (4) (2019) 1356–1372.
[31] M. Bidoki, M.R. Moosavi, M. Fakhrahmad, A semantic approach to extractive multi-document summarization: Applying sentence expansion for tuning of conceptual densities, Inf. Process. Manage. 57 (6) (2020) 102341.
[32] R. Sahba, N. Ebadi, M. Jamshidi, P. Rad, Automatic text summarization using customizable fuzzy features and attention on the context and vocabulary, in: 2018 World Automation Congress (WAC), IEEE, 2018, pp. 1–5.
[33] T. Vetriselvi, N. Gopalan, An improved key term weightage algorithm for text summarization using local context information and fuzzy graph sentence score, J. Ambient Intell. Humaniz. Comput. (2020) 1–10.
[34] P. Verma, H. Om, MCRMR: Maximum coverage and relevancy with minimal redundancy based multi-document summarization, Expert Syst. Appl. 120 (2019) 43–56.
[35] R. Nallapati, F. Zhai, B. Zhou, Summarunner: A recurrent neural network based sequence model for extractive summarization of documents, in: Thirty-First AAAI Conference On Artificial Intelligence, 2017.
[36] W.S. El-Kassas, C.R. Salama, A.A. Rafea, H.K. Mohamed, Automatic text summarization: A comprehensive survey, Expert Syst. Appl. (2020) 113679.
[37] T. Young, D. Hazarika, S. Poria, E. Cambria, Recent trends in deep learning based natural language processing, Ieee Comput. Intell. Mag. 13 (3) (2018) 55–75.
[38] A.M. Rush, S. Chopra, J. Weston, A neural attention model for abstractive sentence summarization, 2015, arXiv preprint arXiv:1509.00685.
[39] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, 2014, arXiv preprint arXiv:1409.0473.
[40] S. Chopra, M. Auli, A.M. Rush, 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, 2016, pp. 93–98.
[41] A. See, P.J. Liu, C.D. Manning, Get to the point: Summarization with pointer-generator networks, 2017, arXiv preprint arXiv:1704.04368.
[42] P. Li, W. Lam, L. Bing, Z. Wang, Deep recurrent generative decoder for abstractive text summarization, 2017, arXiv preprint arXiv:1708.00625.
[43] L. Liu, Y. Lu, M. Yang, Q. Qu, J. Zhu, H. Li, Generative adversarial network for abstractive text summarization, in: Proceedings Of The AAAI Conference On Artificial Intelligence, 32, (1) 2018.
[44] G. Rizzo, T.H.M. Van, Adversarial text generation with context adapted global knowledge and a self-attentive discriminator, Inf. Process. Manage. 57 (6) (2020) 102217.
[45] R. Paulus, C. Xiong, R. Socher, A deep reinforced model for abstractive summarization, 2017, arXiv preprint arXiv:1705.04304.
[46] A. Fan, D. Grangier, M. Auli, Controllable abstractive summarization, 2017, arXiv preprint arXiv:1711.05217.
[47] S. Wang, X. Zhao, B. Li, B. Ge, D. Tang, Integrating extractive and abstractive models for long text summarization, in: 2017 IEEE International Congress On Big Data, BigData Congress, IEEE, 2017, pp. 305–312.
[48] M. Tomer, M. Kumar, Improving text summarization using ensembled approach based on fuzzy with LSTM, Arab. J. Sci. Eng. 45 (12) (2020) 10743–10754.
[49] F. Zaman, M. Shardlow, S.-U. Hassan, N.R. Aljohani, R. Nawaz, HTSS: A novel hybrid text summarisation and simplification architecture, Inf. Process. Manage. 57 (6) (2020) 102351.
[50] M. Yang, Q. Qu, W. Tu, Y. Shen, Z. Zhao, X. Chen, Exploring human-like reading strategy for abstractive text summarization, in: Proceedings Of The AAAI Conference On Artificial Intelligence, 33, (01) 2019, pp. 7362–7369.
[51] M. Yang, C. Li, Y. Shen, Q. Wu, Z. Zhao, X. Chen, Hierarchical human-like deep neural networks for abstractive text summarization, IEEE Trans. Neural Netw. Learn. Syst. (2020).
[52] D. Zajic, B. Dorr, R. Schwartz, Bbn/umd at duc-2004: Topiary, in: Proceedings Of The HLT-NAACL 2004 Document Understanding Workshop, Boston, 2004, pp. 112–119.
[53] R. Nallapati, B. Zhou, C. Gulcehre, B. Xiang, et al., Abstractive text summarization using sequence-to-sequence rnns and beyond, 2016, arXiv preprint arXiv:1602.06023.
[54] Q. Zhou, N. Yang, F. Wei, M. Zhou, Selective encoding for abstractive sentence summarization, 2017, arXiv preprint arXiv:1704.07073.
[55] K. Woodsend, M. Lapata, Automatic generation of story highlights, in: Proceedings Of The 48th Annual Meeting Of The Association For Computational Linguistics, 2010, pp. 565–574.
[56] D. Parveen, H.-M. Ramsl, M. Strube, Topical coherence for graph-based extractive summarization, in: Proceedings Of The 2015 Conference On Empirical Methods In Natural Language Processing, 2015, pp. 1949–1954.
[57] X. Wan, Towards a unified approach to simultaneous single-document and multi-document summarizations, in: Proceedings Of The 23rd International Conference On Computational Linguistics, Coling 2010, 2010, pp. 1137–1145.
[58] J. Cheng, M. Lapata, Neural summarization by extracting sentences and words, 2016, arXiv preprint arXiv:1603.07252.