ISSN 2278-3091
Volume 8, No.6, November – December 2019
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse45862019.pdf
https://doi.org/10.30534/ijatcse/2019/45862019

Survey on Abstractive Text Summarization using various approaches

Arun Krishna Chitturi [1], Saravanakumar Kandasamy [2]
[1] Vellore Institute of Technology, Vellore, India, chitturiarunkrishna@gmail.com
[2]Professor at Vellore Institute of Technology, Vellore, India, ksaravanakumar@vit.ac.in

ABSTRACT

Text summarization is the core aspects of Natural Language processing. Summarized text should consist of unique sentences. It is used in many situations in today's Information technological word, one of the best examples is in understanding customer feedbacks in companies. This job can be done by humans, but if the text or data that has to be summarized then it will consume lot of time and work force. This situation lead to birth of different approaches in summarization. This paper addresses and concentrates on various methods and approaches and their results in abstractive text summarization. This survey gives an insight about different types of text summarization and various methods used in abstractive text summarization in recent developments.

Key words : abstractive summarization, decoder, encoder, multi document summarization

1. INTRODUCTION

Summarization is very well useful to us in today's world. The main aim of abstractive text summarization is to produce shortened version of input text with relevant meaning[7]. The adjective abstractive is utilized because it denotes that the generated summary is not a combination or selection of some repeated sentences, but it a paraphrasing of core contents of the input document [8]. Abstractive summarization is a very difficult problem apart from Machine translation. The main challenge in ATS is to compress the matter of input document in an optimized way so that the main concepts of the document are not missed [8]. In current technologically advancing world, volumes of data is increasing and it is very difficult to read the required data in short time[6]. It is a pretty task to collect the required information and then convert into summarized form. Therefore, text summarization came into demand. Summarized text saves time and helps in avoiding retrieving massive text. Abstractive Text summarization can be combined with numerous intelligent systems on the basis of NLP technologies like information retrieval, question answering, and text classification to find the particular information [9]. If latent structure information of the summaries can be incorporated into abstractive summarization model, then the quality of summaries generated can be improved [10]. In some research works, topic models are used to capture the latent information from the input paragraph or documents. Despite having many hurdles abstractive text summarization faces core issues like (i) Neural sequence-to-sequence models which try to produce generic summary, which include mostly used phrases (ii) The generated summaries are less readable and are not grammatically perfect [11]. Summarization is divided into following types: (a) Extractive text summarization (b) Abstractive text summarization [6]. Extractive summarization extracts the frequently used or only precise phrases without modifying them and generates the summary. Whereas abstractive summarization generates new sentences and also optimally decreases the length of the document. Abstractive is better and qualitative than extractive as it takes data from multiple documents and then generate precise information of summary. Abstractive summarization is again achieved in two ways. They are: (a) Structure based approach (b) Semantic based approach. Neural network models on the basis of encoder decoder for machine translation achieved good ROGUE scores [12]. Abstractive approaches generate summary similar to summary generated by humans but they are more expensive [13]. On the basis of current state of RNN in Attentive RNN the encoder computes score over the input sentences [14]. The main problem in ATS are (a) Long document summarization (b) Abstractive metric (c) Controlling output length. F1 scores are evaluated generally using ROUGE metrics [15]. Recall-Oriented Understudy for Gisting Evaluation (ROUGE) metric was proposed by (Lin, 2004) [24]. Named Entity Recognition is also one of the core application in NLP which helps in removing ambiguity [28]. Information Retrieval is also highly difficult and it requires quality documents[37].

2. SURVEY

2.1 Semantic Link Network For Summarization[1]:

SLN is a semantics self-formulated for semantically organizing resources to support advanced information services like Abstractive Text Summarization [1]. According the author the semantic link network, which is used in Abstractive text summarization, has following important components:

a) SLN construction

(i) Concept extraction with relation identification

(ii) Event extraction with relation identification

- Event trigger extraction
- Event argument extraction
- Event relation extraction

b) Semantic link network summarization

a) Semantic link network construction

SLN construction involves the following two important components

(i)**Concept extraction with relation identification**

Events, concepts are considered as two main roots or base units of information present in the documents. The relation present between concepts carries critical information between events. The main advantage of using concepts and their relations in Semantic link network (SLN) is that few events with indirect relations can be connected easily.

Here the relation between the concepts are nothing but the phrases between the concepts. The event triggers are the one's which are verbs. Some valid syntactic patterns are used to differentiate between the event triggers i.e. the verbs. Some illustrative syntactic patterns which ae used are "be", "be"- Noun Phrase - Preposition, "be"- Adjective - Preposition.

(ii) **Event extraction with relation identification**

Event extraction with relation identification is also called as Frame net based event extraction. Some pre-defined event schemes are present, and with the help of these schemes structured event information is extracted. Many present approaches rely on Automatic Constant Extraction (ACE), which characterizes eight types of events with thirty-three Subtypes in total. Event is a mixture or combination of Automatic Constant Extraction (ACE), which characterizes eight types of events with thirty-three Subtypes in total. Frame net corpus consists of entire interpretations of Semantic frames along with the relation between the semantic frames. Each formal statement is considered as a frame. Many frames together with a hierarchy complete the event schemes. Event extraction consists of following components

- Event trigger extraction

In this step all event triggers are identified and then the event types are also classified. In a given sentence. A log-linear model is utilized in event type classification. In a particular sentence, X = {x1,x2,…xn} with the triggers T = {t1,t2,..tn}, ti denotes ith trigger word, til denotes Lemma.

$P(f \mid ti, X) = [\exp(\Theta T\ g(f,ti,x))] / [\Sigma f \in Fi\ \exp(\Theta T\ g(f,ti,x))]$

- Event argument extraction

In this step, concepts which act as arguments are identified and their argument roles are classified. This constitutes for event trigger extraction.

- Event relation extraction

The structure of the sentences and their features are weighed to ascertain the relation between the events. Some of the common categories of semantic links between the events are Condition link, Sequential link, Attribution link.

b) **Semantic link network summarization**

The summary that is extracted should contain most important events and concepts. It should also be semantically coherent. For achieving this , we increase the saliency scores of selected concepts and events. Let us consider E and C which will be denoting all types of event nodes and concept nodes. Here E is unique concept and C is saliency.

$\Sigma e \in E\ \Theta T\ f(e) + \Sigma c \in C\ \Psi T\ g(c)$

f(e) - Features of Event e

g(c) – Features of Concept c

2.2 Improved Semantic Graph approach For Summarization [2]:

We know that two approaches are there for summarization. They are :

(i)Extractive Text Summarization

(ii) Abstractive Text summarization

For achieving abstractive text summarization, there are 2 approaches, they are

(i)Linguistic approach

(ii) Semantic approach.

Usually the graph based approach requires human intervention and it is also specified or constrained to one domain. It can't be used for other domains. Naïve Bayes is supervised algorithm and is known for it's robustness [48]. But then it also requires human intervention.

The author proposes a semantic graph based method for Multi document abstractive summarization(MDAS). The proposed graph based ranking algorithm is improvised by using Predictive Argument Structure (PAS) semantic similarity and 2 types of semantic relationships. Integrating the semantic similarity will be helpful in determining the relation between PAS and is also helpful in detecting redundancy. This approach has the following main components.

(i)Creation of Semantic graph

a. **Semantic role labelling**

This is the first step and in this stage, each sentence is parsed and Predictive Argument Structure(PAS) is extracted from them. Multiple documents are segmented into bunch of sentences. Now every sentence is given with a key, which is based on location and time of the sentence. A SENNA which is a semantic role parser is utilized to perform semantic text analysis in Abstractive Text Summarization. It also decides PAS from sentence by labelling semantic phrases or also called semantic arguments. These semantic arguments are classified as (a) Core arguments (b) Adjunctive arguments.

b. **Semantic similarity matrix**

Now in this stage, semantic similarity scores of PAS are calculated in pairs. Based on these semantic similarity scores a matrix is built. Verb, Location, Noun and Time arguments of each PAS are differentiated or related with other PAS to find out pair wise similarities.

First Jiang's measure finds semantic distance of the concepts.

Jiangdist (C1, C2) = IC (C1) + IC (C2) − 2 x IC (lso (C1,C2))

IC is Information content of a concept

lso is Least Common subsumer

Jiang's measure uses to find lso, IC. IC is calculated by

IC (C) = − log P(C)

P(C) **-** extension of concept C

Semantic similarity of any 2 PAS's is calculated by

Sim sem (pi , pj) = Sim verb (pi , pj) + Sim arg (pi , pj) + Sim tmp (pi , pj) + Sim loc (pi , pj)

Sim verb (pi , pj) **-** predicates similarity

Sim arg (pi , pj) **-** sum of semantic arguments similarities from corresponding PASs

Sim tmp (pi , pj) **-** Similarity of time arguments

Sim loc (pi , pj) **-** similarity of location arguments

c. **Semantic graph**

Semantic graph is created from similarity matrix, if similarity weight Sim (pi , pj) between PAS's pi and pj is greater than zero then link is set up between PAS, else link is not setup. The link represents similarities weight between PAS.

(ii) **Improved weighted graph based working algorithm**

Present prevailing graph based methods utilize procedures which are similar to page ranking algorithm. They also utilize content similarity apart from semantic similarity to determine relations between sentences. The proposed model utilizes an improved ranking algorithm based on weighted graph (IWGRA). IWGRA uses edge weights in analysis of vertices (PAS). The edge weight is found from PAS to PAS similarity. The importance score of vertex IWGR(vi) is calculated by:

IWGR(vi) = (1 − dp) + dp. Σvj ∈ In(vi) [IWGR(vi) . Wji]/ [Σvz ∈ Out(vi) Wzj]

dp – damping factor which is considered as 0.85 in the ranking model

In(vi) – Denotes vertices that point to a vertex Vi

Out(vj) – Set of links going out of vertex vj

Wij – Indicates edge between vertices Vi and Vj

Wzj – Weights of outgoing links from Vj

(iii) **Maximum Marginal Reference(MMR)**

There is a possibility that the ranking algorithm may give identical rank score to PAS representing identical content finally leading to redundancy in the summary. MMR is utilized to reduce the redundancy.

MMR = argmax pi ∈ R/P [α, RS(pi)] – (1- α) max pj∈P Sim(pi,pj)

R – Set of predicate argument structures

P – Set of already chosen PAS

RS(Pi) – Ranking score of PAS

α – Tuning parameter between PAS which is 0.6 for optimal performance

2.3 Using Adversarially Regularized Autoencoders [3]:

In today's world, many industries or companies utilize abstractive text summarization to understand the customer feedbacks. The author proposed abstractive summarization which can be trained unsupervisedly. This method is dependent on Adversarially Regularized Auto Encoder (ARAE) model. Along with that Conditional Adversarially Regularized Autoencoder (CARAE) is proposed. CARAE is an extension of condition nodes to ARAE because the additional information regarding cluster will be helpful in summarization. Initially review data given by the customers is summarized. This summarized data is gathered on the basis of topics. Now summarization is conducted on peer review data in Korean and Opinosis data set which is in English language.

Many existing models utilize autoencoders in order to extract summaries. In the proposed model code vectors of ARAE are utilized to summarize review data classified by topic. [23] Zhao et al. (2017) proposed ARAE and CARAE algorithms. These both training algorithms are identical to each other. Autoencoder is utilized to do abstractive summarization over the encoded code space which represented each sentence. Despite summarizing the data by ranking individual set of lines, summarization is done by utilizing individual line code space. In the proposed method embedded vectors of ARAE are used.

CARAE model consists of following components:

- Input
- Decoder
- Encoder
- Critic
- Generator

Min φ,ψ,θ Lrec (φ, ψ) + λ W(Pr, Pg)

Lrec (φ, ψ) is the reconstruction loss of the autoencoder and consists of encoder parameters φ and decoder parameters ψ. [16] The training algorithm for both ARAE and CARAE is identical and it was proposed by Zhao et al. (2017). The training algorithm consists of the following steps:

a. Train the autoencoder for reconstruction (Lrec(φ, ψ))

b. Train the critic (Lcri(ω)) (Repeat k times)

c. Train the encoder and generator adversarially to critic (Lencs(φ, θ))

The process carried out in summarization is as follows

First Topic 'Y' (ky review data) is passed through the encoder. After it is passed through the encoder ky code vectors are generated. The average of encoded code vectors of individual clusters is found, resulting in average code vector 'cy'. Cluster topic information is generated and passed through the decoder. Finally the decoder generates the summarized text.

cˆy = 1/Ky (Σ Ky k=1 ck)

x = vn

hj = RNN(xj , hj−1; φ)

encφ(x) = hn = c

h˜ j = RNN(xj , h˜ j−1, c; ψ)

$p\psi(x|c) = \pi n\ j=1$ softmax $(Wh\tilde{}\ j + b)xj$

2.4 Using LSTM-CNN for ATS [4]:

Over the previous few years, the major part of work was done on extractive text summarization [28, 29,30,31,32,33,34]. Convolution neural networks are used widely. They are used in sentiment analysis which is also a core issue in Natural language processing [35]. Abstractive text summarization is very tedious task. The author uses LSTM-CNN based ATS (Abstractive Text Summarization) structure (ATSDL) which builds new sentences in the summary. ASTDL consists of two main stages. They are:

- Extraction of phases from source text
- Generation of summarized text using Deep learning

The main advantage of Extractive text summarization models is that the sentences summarized in the models match the requirements of syntactic nature. On the contrast, the disadvantage of extractive text summarization model is that the summarized text isn't semantically consistent. Perhaps the key reason is that sentences which are adjacent inside summary, need not be compulsorily be side by side in the source text.

The procedure of the proposed model is mentioned below:

a. Firstly, the LSTM (long short-term memory) and the CNN model is fused together to increase the performance of the summarization.

b. ATSDL (Abstractive Text Summarization Deep Learning) utilizes phrase extraction for obtaining the important phrases from the sentences. Now LSTM comes into scene, ASTDL utilizes LSTM to learn allocation of sentences. Now training of the model is done. After the training this model generates flow of phrases, resulting in summarized text.

c. Incase of rare words, phrase location information is utilized to clear the limited words issue. Therefore more natural words are generated.

The main perspectives of the proposed of LSTM-CNN model are:

a. **Input & Output**

Despite of taking words as input, phrases are taken as input and output sequences. By doing so the model can produce more meaningful and semantical sentences. [26] Phrases are key concepts in field of phrase extraction. They also play an important role in phrase filing [27]. The phrases are divided as subject, object and relational one's.

Relational phrases gives an insight on relational information in the sentence, for instance 'is', 'won award for' and 'invention of'. Each and every relational phrase has two entities. They are, subject phrase and object phrase. Subject/object phrases consists of nouns, adjectives and objects which are in relation with relation phrase, such as 'We', 'Arun'.

b. **Convolutional Phrase Encoder**

The main objective behind the author's choice of choosing this CNN. They are

- A single layer CNN can be trained efficiently despite of any long-term dependencies in the model.
- CNN gives effective output in sentiment analysis.

d - dimension of word embedding

s - document phrase consisting of a sequence of n words

W – dense column matrix $W \in K(n*d)$

Next a temporal narrow convolution is applied between W and Kernel K of width c

$fji = \tanh(Wj{:}j+c-1 \otimes K + b)$

$\otimes$ is the Hadamard Product followed by a sum over all elements. f i j denotes the j-th element of the i-th feature map f i and b is the bias.

c. **Recurrent Document Encoder**

Apart from RNN gated recurrent units (GRU) perform well and also posses good architectures. Despite of GRU having an upper hand in training time, the proposed model utilizes LSTM because it is is simple to change the parameters and has stronger value of success theoritically. While reading the input phrases the forward propagation of LSTM is computed as follows:

$ft = \sigma\ (Wf * [ht\text{-}1\ ,\ xt\ ] + bf)$

$it = \sigma\ (Wf * [ht\text{-}1\ ,\ xt\ ] + bi)$

$Ct = \tanh\ (Wc * [ht\text{-}1\ ,\ xt\ ] + bc)$

$Ct = ft * Ct\text{-}1 + it * Ct$

$Ot = \sigma\ (Wf * [ht\text{-}1\ ,\ xt\ ] + bo)$

$ht = Ot * \tanh(Ct)$

2.5 Using Neural networks along with Encoders [5]:

We know that Neural networks are broadly utilized in NLP because of their performance. Neural networks based encoder decoder models are utilized in neural machine translations [36], speech recognition [38], image captioning[39], ATS. The encoder gathers the entire input sequence and generates a static dimensions feature vector followed by decoder uses feature representation for generating the required output. The main problems in encoder and decoder models are:

- A fixed target vocabulary is utilized by the decoder to generate probability distribution at every step, this leads to unknown words problem or lack of vocabulary problem.
- Decoder and encoder model usually generate repeated phrases

Now a dual encoding model for Abstractive text summarization (DEATS) is proposed to solve these problems. DEATS is an extension to existing sequence framework. The proposed DEATS has following main components:

- Primary encoder
- Secondary encoder
- Decoder

Secondary encoder is dependent on the input and previous output. This produces a new context vector as an input to decoder. This generated context vector makes decoder to get more meaningful information. Finally leading to a better output. A multistep decoding is done to avoid repetition.

For every iterator the Primary encoder starts reading and produces hidden representation (hp) and Content representation (cp) for entire text. The decoder produces partial fixed length sequence for every ‘k” decoding steps which is modeled as current decoded content representation (cd). The secondary encoder fulfils finer encoding on input sequence for every ‘k” decoding steps.

a. **Primary encoder**

Primary encoder generates coarse encoding by utilizing GRU-based RNN. The main reason for using a GRU-based RNN is that it can effectively capture dependencies of different time scales as mentioned below.

$ut = \sigma (Wu[xt, ht-1])$

$rt = \sigma (Wr [xt, ht-1])$

$ht = \tanh(Wh [xt,rt \circ ht-1)])$

Wu, Wr, and Wh are parameter matrices.

xt - input embedding vector

ht - hidden state vector

Bi-GRU comprises of backward GRU. It gives hidden state representation to each word.

b. **Secondary encoder**

The primary encoder reads the input statement and gives hidden representation. On contrast, the secondary one is constructed with unidirectional GRU RNN. This accepts the input sequence at every K decoding steps. In parallel, the importance weight αt is computed.

c. **Decoder**

Attention mechanism, copy mechanism, pointer-generator network and coverage mechanism are used in the basic sequence-to-sequence model to achieve better performance.

3. EVALUATION METHODOLOGIES

ROUGE – 1 is referred as R - 1
ROUGE – 2 is referred as R - 2
ROUGE – L is referred as R – L

3.1 Semantic Link Network For Summarization[1]:

Now, for the purpose of evaluation, tests are performed on benchmark DUC 2005, DUC 2006 and DUC 2007 datasets[1]. DUC 2005 consists of 50 topics, DUC 2006 consists of 50 topics, DUC 2007 consists of 45 topics. These datasets consists of 250 words. ROUGE1, ROUGE2 ROUGE-SU4 are used as metrics. And SLN performed well as the TextRank, LexPageRank, NIST Baseline, MedianDUC results were better than baseline statistics. For instance, the results of experiments on DUC 2006 are as follows, ROGUE-1 score for Symantec role labelling (SRL) for other models is 0.38158 but for the proposed model it is 0.39017. ROGUE-2 score for Symantec role labelling (SRL) for other models is 0.07398 but for the proposed model it is 0.11033. ROGUE-SU4 score for Symantec role labelling (SRL) for other models is 0.13001 but for the proposed model it is 0.14844. Similarly the proposed system was dominant than that of other systems.

3.2 Improved Semantic Graph approach For Summarization [2]:

ROUGE and Pyramid metrics are used for evaluation of the proposed system. Here proposed semantic approach is compared to latest abstractive approach for multi-document summarization. The experiments were performed on DUC 2002 dataset as it is considered as benchmark dataset. Proposed Sem-graph-both-rel gave the Mean coverage score of 0.5441 and AVG-precision of 0.7563 and AVG-F-Measure of 0.6396. Best (System code:19) gave the Mean coverage score of 0.2783 and AVG-precision of 0.7452 and Average-F-Measure of 0.4053 [17]. Proposed Sem-graph-both-rel gave the R - 1 score of 0.417 and R - 2 of 0.108. Event graph model gave the R - 1 score of 0.415 and R - 2 of 0.116. DUC-2002 Best (system 21) gave the R - 1 score of 0.395 and R - 2 of 0.103 [18].

3.3 Using Adversarially Regularized Autoencoders [3]:

R - 1, R - 2 and R - L are used for evaluation by the author on Korean and Opinosis data set, which is in English[21]. ARAE model gave R - 1, R - 2, R - L scores of 28.94, 8.43, 28.26 respectively[20]. Proposed CARAE model gave R - 1, R - 2, R - L scores of 34.29, 7.53 and 33.74 respectively. Opinosis model gave R - 1, R - 2 scores of 28.31 and 8.53 respectively. AE and CAE models are trained using Adam with a learning rate of 1. Scores of AE model were 26.27, 11.13 and 26.16 respectively [22]. Scores of CAE model were 25.10, 10.26 and 24.71 respectively. BLEU is also used for evaluation. BLEU scores of AE, CAE, ARAE and CARAE are 35.90, 39.84, 44.00 and 50.67 respectively [25].

3.4 Using LSTM-CNN for ATS [4]:

The results depicted from the experiments on CNN and DailyMail datasets shows that ATSDL framework surpasses the other models in semantic nature and also in syntactic nature. Proposed ASTDL gave the R - 1, R - 2 scores of 34.9 and 17.8. Moses is one of the statistical approaches to machine translation. It considers parallel data and utilizes co-occurrence of words and phrases. MOSES gave R - 1, R - 2 scores of 27.8 and 14.1.

3.5 Using Neural networks along with Encoders [5]:

The author uses CNN/DAILY MAIL dataset to evaluate and R - 1, R - 2 and R - L scores. Seq2seq+atten model gave R - 1, R - 2 and R - L scores of 31.34, 11.79, 2810 respectively [40]. Words-lvt2k-temp-att model gave R - 1, R - 2 and R - L scores of 35.46, 13.30 and 32.65 respectively [41]. Summa-Runner-abs model [42] gave R - 1, R - 2 and R - L scores of 37.40, 14.50 and 33.40 respectively. Pointer-generator model [43] gave R - 1, R - 2 and R - L scores of 36.44, 15.66 and 33.42 respectively. RL+ML model[44] gave R - 1, R - 2 and R - L scores of 39.87, 15.82 and 36.90. Proposed DEATS model gave R - 1, R - 2 and R - L scores of 40.85, 18.08 and 37.13 respectively. When TOPIARY method is used on DUC 2004 dataset it gave R - 1, R - 2 and R - L scores of 25.12, 6.46, 20.12 respectively. SEASS gave scores of 29.12, 9.56, 25.51. DEATS gave scores of 29.91, 9.61 and 25.95.

4. FINDINGS

Table 1 shows the datasets used in all the surveyed papers. The source of the datasets is also mentioned.

Table 1:. Datasets

Papers	Dataset used	Data sources used
[1]	Document Understanding Conference dataset 2005 (DUC 2005) Document Understanding Conference dataset 2006 (DUC 2006) Document Understanding Conference dataset 2007 (DUC 2007)	www-nlpir.nist.gov
[2]	Document Understanding Conference dataset (DUC-2002)	www-nlpir.nist.gov
[3]	Korean and Opinosis data set Opinosis data set English data set.	[51]
[4]	CNN dataset and DailyMail dataset	1.[45] 2.[46] 3.[47]
[5]	1)CNN/DailyMail dataset 2) DUC 2004 dataset	1.[52] 2.[53]

Table 2 shows the results and scope along with the applications of the papers [1] [2] [3] [4] [5] in the respective order.

Table 2: The results table

Results	Proposed work	Applications
Proposed SLN model performed well as Textrank, Lexpagerank, NIST Baseline results were better than baseline statistics	Author proposed AMDS approach which changes documents into a Semantic Link Network of concepts and events, and finally to summary by using SLN	Practically suitable for document summarization
Here proposed semantic approach is compared to latest abstractive approach for multi-document summarization. Semantic approach had better results	Proposes a semantic graphic-based MDAS technique. The main objective is to overcome existing disadvantages of existing graphic methods.	Suitable for Abstractive Summarization
Proposed CARAE model gave ROUGE-1, ROUGE-2, ROUGE-L scores of 34.29, 7.53 and 33.74 respectively which is better than Opinosis model and AE models	Proposed ARAE model with better learning with GAN and CARAE model	Used in Abstractive Summarization
The results depicted from the experiments on CNN and DailyMail datasets shows that ATSDL framework surpasses the other models in semantic nature and also in syntactic nature.	Can be used as a reference for developing a new framework or method or approach on abstractive summarization using deep learning	Can be used for abstractive summarization
Proposed dual encoding model gave better results than seq2seq+atten model and words-lvt2k-temp-att model and other models	To train the model by using a hybrid training objective with reinforce-ment learning training and maximum likelihood training.	Can be used for abstractive summarization

Table 3 shows the algorithms used along with whether they are supervised or unsupervised, and the metrics used for evaluation of the papers [1] [2] [3] [4] [5] in the respective order.

Table 3: The algorithms used

Existing algorithms	Supervised/ Unsupervised	Evaluation Metrics
1)Probabilistic event extraction algorithm 2)Multi-modality manifold ranking algorithm 3)TextRank algorithm	Unsupervised	ROUGE1, ROUGE2 ROUGE-SU4
Genetic algorithm Porter stemming algorithm Edit distance algorithm	Unsupervised	ROUGE and Pyramid
1)GAN (Generative adversarial net) algorithm 2)Greedy algorithm 3)Graph-based algorithms 4)Heuristic algorithm.	Unsupervised	ROUGE-1, ROUGE-2 and ROUGE-L
Not stated	Unsupervised	ROUGE-1, ROUGE-2
sequence-to-sequence framework 2) Repetition avoidance mechanism (RAM)	Unsupervised	ROUGE-1, ROUGE-2 and ROUGE-L

5. CONCLUSION

In this paper a survey of how effectively abstractive summarization can be achieved by using different approaches and different models is clearly represented. All the five main models, which are discussed above in the paper, are developed recently in the past one or two years. Apart from these models some other models were also considered for evaluation purposes and differentiation purpose. This paper can be helpful as a tool of reference for the people who are novice in NLP especially constrained to abstractive text summarization. In the future the proposed architecture can be develop and evaluated with more effective metrics for better results in abstractive text summarization.

REFERENCES

1. Li, W., & Zhuge, H. **Abstractive Multi-Document Summarization based on Semantic Link Network**. *IEEE Transactions on Knowledge and Data Engineering*, 2019. https://doi.org/10.1109/TKDE.2019.2922957
2. Khan, A., Salim, N., Farman, H., Khan, M., Jan, B., Ahmad, A., & Paul, A.. **Abstractive text summarization based on improved semantic graph approach**. *International Journal of Parallel Programming*, 46(5), 992-1016, 2018. https://doi.org/10.1007/s10766-018-0560-3
3. Kong, H., & Kim, W. **Generating summary sentences using Adversarially Regularized Autoencoders with conditional context**. *Expert Systems with Applications*, 130, 1-11, 2019.
4. Song, S., Huang, H., & Ruan, T. **Abstractive text summarization using LSTM-CNN based deep learning**. Multimedia Tools and Applications, 78(1), 857-87,2019. https://doi.org/10.1007/s11042-018-5749-3
5. Yao, K., Zhang, L., Du, D., Luo, T., Tao, L., & Wu, Y. **Dual encoding for abstractive text summarization.** *IEEE transactions on cybernetics*,2018. https://doi.org/10.1109/TCYB.2018.2876317
6. Moratanch, N., & Chitrakala, S. **A survey on abstractive text summarization.** *In 2016 International Conference on Circuit, power and computing technologies (ICCPCT) (pp. 1-7). IEEE.* https://doi.org/10.1109/ICCPCT.2016.7530193
7. Rush, A. M., Chopra, S., & Weston, J. **A neural attention model for abstractive sentence summarization**. *arXiv preprint arXiv:1509.00685*, 2015. https://doi.org/10.18653/v1/D15-1044
8. Nallapati, R., Zhou, B., Gulcehre, C., & Xiang, B. **Abstractive text summarization using sequence-to-sequence rnns and beyond**. *arXiv preprint arXiv:1602.06023,* 2016. https://doi.org/10.18653/v1/K16-1028
9. Moawad, I. F., & Aref, M. **Semantic graph reduction approach for abstractive Text Summarization**. In 2012 *Seventh International Conference on Computer Engineering & Systems (ICCES)* (pp. 132-138). IEEE. 2012, November. https://doi.org/10.1109/ICCES.2012.6408498
10. Li, P., Lam, W., Bing, L., & Wang, Z. **Deep recurrent generative decoder for abstractive text summarization.** *arXiv preprint arXiv:1708.00625*, 2017. https://doi.org/10.18653/v1/D17-1222
11. Liu, L., Lu, Y., Yang, M., Qu, Q., Zhu, J., & Li, H. **Generative adversarial network for abstractive text summarization.** *In Thirty-second AAAI conference on artificial intelligence*. 2018, April.
12. Paulus, R., Xiong, C., & Socher, R. **A deep reinforced model for abstractive summarization**. *arXiv preprint arXiv:1705.04304*, 2017.
13. Dalal, V., & Malik, L. **A survey of extractive and abstractive text summarization techniques**. *In 2013 6th International Conference on Emerging Trends in*

Engineering and Technology (pp. 109-110). IEEE, 2013, December.
https://doi.org/10.1109/ICETET.2013.31

14. Chopra, S., Auli, M., & Rush, A. M. (2016, June). **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), 2013, December. https://doi.org/10.18653/v1/N16-1012
15. 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, August
16. Zhao, Z., Dua, D., & Singh, S. (2017). **Generating natural adversarial examples**. *arXiv preprint arXiv:1710.11342,* 2010, August.
17. Harabagiu, S. M., & Lacatusu, F. **Generating single and multi-document summaries with gistexter.** *In Document Understanding Conferences* (pp. 11-12), 2002, July.
18. Glavaš, G., & Šnajder, J. **Event graphs for information retrieval and multi-document summarization.** *Expert systems with applications*, 41(15), 6904-6916, 2014. https://doi.org/10.1016/j.eswa.2014.04.004
19. Mihalcea, R., Corley, C., & Strapparava, C. **Corpus-based and knowledge-based measures of text semantic similarity.** In Aaai (Vol. 6, No. 2006, pp. 775-780), 2006, July.
20. Pan, S., Hu, R., Long, G., Jiang, J., Yao, L., & Zhang, C. **Adversarially regularized graph autoencoder for graph embedding**. *arXiv preprint arXiv:1802.04407*, 2018. https://doi.org/10.24963/ijcai.2018/362
21. 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, August.
22. Kingma, D. P., & Ba, J. **Adam: A method for stochastic optimization**. *arXiv preprint arXiv:1412.6980.* 2014.
23. Zhao, Z., Dua, D., & Singh, S. **Generating natural adversarial examples.** *arXiv preprint arXiv:1710.11342*. 2017.
24. Lin, C. Y. **Rouge: A package for automatic evaluation of summaries.** *In Text summarization branches out* (pp. 74-81). 2004.
25. Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. **BLEU: a method for automatic evaluation of machine translation.** *In Proceedings of the 40th annual meeting on association for computational linguistics* (pp. 311-318). Association for Computational Linguistics. 2002, July.
26. Angeli, G., Tibshirani, J., Wu, J., & Manning, C. D. **Combining distant and partial supervision for relation extraction.** *In Proceedings of the 2014 conference on empirical methods in natural language processing* (EMNLP) (pp. 1556-1567). 2014, October. https://doi.org/10.3115/v1/D14-1164
27. Bing, L., Li, P., Liao, Y., Lam, W., Guo, W., & Passonneau, R. J. **Abstractive multi-document summarization via phrase selection and merging**. *arXiv preprint arXiv:1506.01597.* 2015. https://doi.org/10.3115/v1/P15-1153
28. Kandasamy, S., & Cherukuri, A. K. (2018). **Query expansion using named entity disambiguation for a question□answering system**. *Concurrency and Computation: Practice and Experience*, e5119.
29. Cheng, J., & Lapata, M. **Neural summarization by extracting sentences and words**. *arXiv preprint arXiv:1603.07252.* 2016
30. Song, S., Huang, H., & Ruan, T. **Abstractive text summarization using LSTM-CNN based deep learning.** *Multimedia Tools and Applications*, 78(1), 857-875. 2019. https://doi.org/10.1007/s11042-018-5749-3
31. Erkan, G., & Radev, D. R. Lexrank**: Graph-based lexical centrality as salience in text summarization**. *Journal of artificial intelligence research*, 22, 457-479. 2004 https://doi.org/10.1613/jair.1523
32. Litvak, M., & Last, M.. **Graph-based keyword extraction for single-document summarization**. *In Proceedings of the workshop on Multi-source Multilingual Information Extraction and Summarization* (pp. 17-24). Association for Computational Linguistics. 2008, August. https://doi.org/10.3115/1613172.1613178
33. Ribeiro, R., Marujo, L., Martins de Matos, D., Neto, J. P., Gershman, A., & Carbonell, J. **Self reinforcement for important passage retrieval.** *In Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval* (pp. 845-848). ACM. 2013, July. https://doi.org/10.1145/2484028.2484134
34. Riedhammer, K., Favre, B., & Hakkani-Tür, D. **Long story short–global unsupervised models for keyphrase based meeting summarization.** *Speech Communication, 52(10), 801-815.* 2010. https://doi.org/10.1016/j.specom.2010.06.002
35. Dr.Balika.J.Chelliah, Darshan Lathia, Sandeep Yadav, Meet Trivedi & Shubham Sagar Soni. **Sentiment Analysis of Twitter Data using CNN.** *International Journal of Emerging Technologies in Engineering Research (IJETER) Volume 6, Issue 4, April (2018)*
36. Sutskever, I., Vinyals, O., & Le, Q. V. **Sequence to sequence learning with neural networks.** *In Advances in neural information processing systems* (pp. 3104-3112). 2014.

37. Kandasamy, S. A. R. A. V. A. N. A. K. U. M. A. R., & Cherukuri, A. K. (2017). **Information retrieval for question answering system using knowledge based query reconstruction by adapted Lesk and latent semantic analysis.** *International Journal of Computer Science & Applications, 14(2).*
38. Biagetti, G., Crippa, P., Falaschetti, L., Orcioni, S., & Turchetti, C. **An investigation on the accuracy of truncated DKLT representation for speaker identification with short sequences of speech frames**. *IEEE transactions on cybernetics*, 47(12), 4235-4249. 2016. https://doi.org/10.1109/TCYB.2016.2603146
39. Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. **Show and tell: A neural image caption generator**. *In Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3156-3164). 2015. https://doi.org/10.1109/CVPR.2015.7298935
40. Bahdanau, D., Cho, K., & Bengio, Y. **Neural machine translation by jointly learning to align and translate**. *arXiv preprint arXiv:1409.0473.* 2014.
41. Nallapati, R., Zhou, B., Gulcehre, C., & Xiang, B. **Abstractive text summarization using sequence-to-sequence rnns and beyond**. *arXiv preprint arXiv:1602.06023.* 2016.
42. 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, February.
43. See, A., Liu, P. J., & Manning, C. D. **Get to the point: Summarization with pointer-generator networks**. *arXiv preprint arXiv:1704.04368*. 2017 https://doi.org/10.18653/v1/P17-1099
44. Paulus, R., Xiong, C., & Socher, R. **A deep reinforced model for abstractive summarization**. *arXiv preprint arXiv:1705.04304*. 2017
45. Sutskever, I., Vinyals, O., & Le, Q. V. **Sequence to sequence learning with neural networks.** *In Advances in neural information processing systems* (pp. 3104-3112). 2014
46. Dunn, M., Sagun, L., Higgins, M., Guney, V. U., Cirik, V., & Cho, K. **Searchqa: A new q&a dataset augmented with context from a search engine.** *arXiv preprint arXiv:1704.05179.* 2017.
47. Grusky, M., Naaman, M., & Artzi, Y. **Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies.** arXiv preprint arXiv:1804.11283, 2018. https://doi.org/10.18653/v1/N18-1065
48. R. B. Saroo Raj, Gurpartap Singh, Balaji S & K. H. Ajit Baskar. **A Model to Predict Loan Defaulters using Machine Learning.** *International Journal of Emerging Technologies in Engineering Research, Volume 6, Issue 10, October (2018)*