Proceedings of the Sixth International Conference on Inventive Computation Technologies [ICICT 2021]
IEEE Xplore Part Number: CFP21F70-ART; ISBN: 978-1-7281-8501-9

Natural Language Processing (NLP) based Text Summarization - A Survey

Ishitva Awasthi
Information Technology
SVKM's NMIMS, MPSTME
Shirpur, India
ishitvaaawasthi@gmail.com

Kuntal Gupta
Information Technology
SVKM's NMIMS, MPSTME
Shirpur, India
kuntal.gupta262@gmail.com

Prabjot Singh Bhogal
Information Technology
SVKM's NMIMS, MPSTME
Shirpur, India
bhogalprabjot1@gmail.com

Sahejpreet Singh Anand
Information Technology
SVKM's NMIMS, MPSTME
Shirpur, India
sahej99@gmail.com

Prof. Piyush Kumar Soni
Department of Information Technology
SVKM's NMIMS, MPSTME
Shirpur, India
piyushkumar.soni@nmims.edu

2021 6th International Conference on Inventive Computation Technologies (ICICT) | 978-1-7281-8501-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICICT50816.2021.9358703

***Abstract*— The size of data on the Internet has risen in an exponential manner over the past decade. Thus, the need for a solution emerges, that transforms this vast raw information into useful information which a human brain can understand. One such common technique in research that helps in dealing with enormous data is text summarization. Automatic summarization is a renowned approach which is used to reduce a document to its main ideas. It operates by preserving substantial information by creating a shortened version of the text. Text Summarization is categorized into Extractive and Abstractive methods. Extractive methods of summarization minimize the burden of summarization by choosing from the actual text a subset of sentences that are relevant. Although there are a ton of methods, researchers specializing in Natural Language Processing (NLP) are particularly drawn to extractive methods. Based on linguistic and statistical characteristics, the implications of sentences are calculated. A study of extractive and abstract methods for summarizing texts has been made in this paper. This paper also analyses above mentioned methods which yields a less repetitive and a more concentrated summary.**

Keywords—Text summarization, extractive, abstractive, reinforcement learning, supervised, unsupervised.

I. INTRODUCTION

[1]The summary of large texts remains an open problem in natural language processing. Automatic Text Summarization is used to summarize large documents. Text summarization is the process of shortening a text document with software, in order to create a summary with the major points of the original document. Text summarization is the technique of reducing a text document with the use of software, in order to create a summary or abstract of the original document. Summarization is done to highlight the important parts of the text.

Text summarizer can be classified based on input type: Single Document, where the input is small in textual context. Basic summarization models are built for such cases. Multi document, where the input can be comparatively long. The complexity increases here as more text leads to more semantic links being generated.

Based on the aim, summarizer can be classified as Generic, where the model treats the input without any bias and prior knowledge. Domain-specific, where the model uses domain information to form a more accurate summary based on known facts. Query-based, where the summary only contains known answers to natural language questions about the input text.

Based on output type summarizer can be classified as: Extractive, where important sentences are selected from the input text to form a summary. Abstractive, where the model forms its own phrases and sentences to offer a more coherent summary, like what a human would generate. In general, creating abstract summaries is a more complex task than extractive methods. Therefore, they are still far from reaching the human level, except for recent advances in the use of neural networks promoted by the advances of neural machine translation and sequencing models.

Applications of text summarizer are media monitoring, search marketing, internal document workflow, financial research, social media marketing, helping disabled people and more.

II. METHODOLOGIES

In this segment, we are about to review numerous outstanding works that have been accomplished on Text Summarization as shown in Fig. 1. We are essentially going to represent their approach and workflow.

A. Extractive

1) Unsupervised

Extractive Unsupervised summarization technique means creating the summary from the given document without using any previous labelled group or classification. There are three ways to do so, firstly graph based, secondly latent variable and lastly, term frequency. These are easy to implement and give satisfactory results. Some of the research done is mentioned below.

Hernández et al., [17] presented a solution of using K-Means Clustering for choosing sentences in extractive text

summarization which is a major disadvantage. The first step is to eliminate stop words, hyphens and redundant white spaces. This is called pre processing of the input text. The next step is to select the feature using n-gram and finding out the weights using Boolean Weighting(BOOL), Term Frequency(TF), Inverse Document Frequency(IDF) or TF-IDF. The next step is to apply KMeans for sentence clustering. KMeans is an iterative process in which values are plotted to the nearest centroid(mean of all values) and then calculating new centroids. In the proposed method, the first sentence is considered as a baseline and the similarity between the sentences is plotted using Euclidean's distance.After the clustering is done using K clusters, the sentences (also called most representative sentences) nearest to the centroids are selected. The proposed method obtains more favourable results than other state-of-art methods.

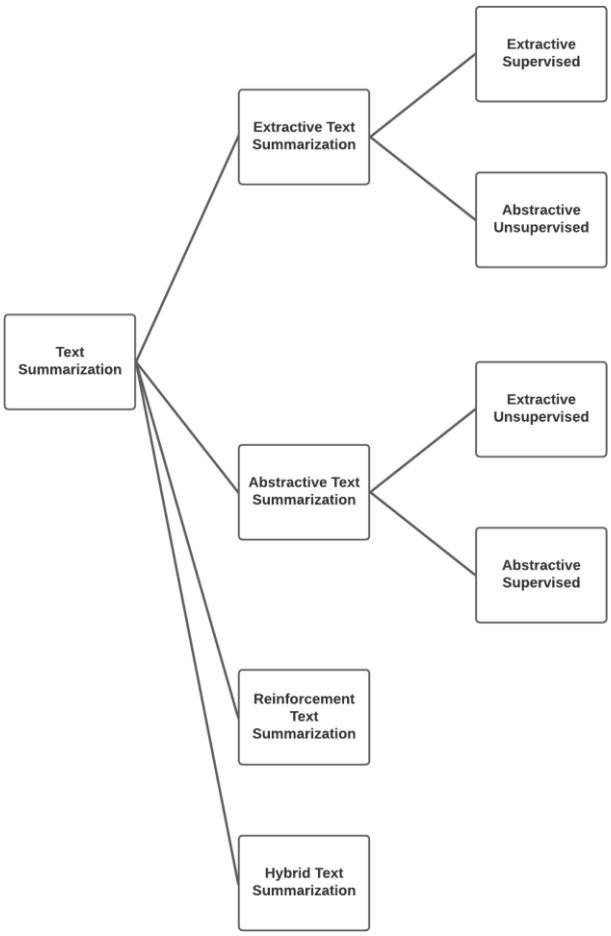

Fig. 1. Taxonomy for Text summarization

Joshi et al., [18] suggested an unsupervised framework for extractive text summarization of a single document called SummCoder. In SummCoder, after the pre-processing, the sentences are converted to fixed length vectors using the skip thought model. For generating summary, sentence selection is done considering 3 scores: Sentence Content Relevance Metric ($score^{ContR}$), Sentence Novelty Metric ($score^{Nov}$) and Sentence Position Relevance Metric ($score^{PosR}$). After calculating all the scores a final score and relative score is calculated. Finally, the summary can be generated by firstly, according to the descending order of relative rank and secondly, according to their occurrence in the input text.

El-Kassas et al., [19] presented a single document and graph-based extractive system called EdgeSumm**.** In the proposed method, firstly pre-processing and lemmatization is done. After that for text representation, a graph is created with the nouns as nodes and non-noun words as the edges. There are "S#" and "E#" node to indicate the starting and ending of the sentence. For each node weight is calculated by counting the frequency of its occurrence. For selecting sentences, there is an assumption that all nouns represent different topics. Firstly, it searches for the most frequent words or phrases and creates a list of the selected nodes and edges. For the source and destination node to be selected then the score must be greater than the average score of all the nodes and to select an edge both the source and destination node must be selected. If the candidate summary(summary generated using the algorithm) exceeds the user-limit then, the sentences in the candidate summary are scored and ranked in ascending order. After that KMeans clustering is applied to group similar sentences and the sentences with higher rank from each cluster are selected to generate the final summary.

Zheng & Lapata, [20] proposed Position-Augmented Centrality based Summarization(PacSumm). It uses graph based ranking algorithms where sentences are the nodes and the edges show the relationship between the nodes.For mapping the sentences, Bidirectional Encoder Representations from Transformers(BERT) was used. There are 2 tasks for pre-training BERT, first task is masked language modelling in which a token is given to the sentence in view of the left and right sentences and second, sentence prediction in which the relationship between two sentences is predicted. For fine tuning BERT, five negative samples per positive sample are given. After finding the representations of all sentences, a pairwise dot product is taken to create an unnormalized matrix. Using this matrix, the sentences will be selected.

Vanetik et al., [21] suggested a Weighted Compression Model for extracting important information from the text. In the proposed model, this is done by shortening the sentence by iteratively removing Elementary Discourse Units(EDUs). Firstly, every term is given a non-negative weight. The weights are assigned using Extractive models of Gillick and Favre [34] and McDonald [35]. The next step is selecting and removing EDUs. A list of EDUs is created using constituency-based syntax trees. From the list, the EDUs that can make a sentence grammatically incorrect if removed are omitted. All the others are removed and the weight of the "important" EDUs are calculated and sorted. For the generating the summary, the EDUs are selected on the basis that weight to cost ratio is maximum and summary length is not exceeded.

Ozsoy & Alpaslan, [22] presented Latent Semantic Analysis(LSA) for text summarization. It is an algebraic-statical method for finding out hidden logical patterns between words and sentences. For text representation, an input matrix is created in which the rows represent the words and the columns represent the sentences. The cells show the TF-IDF value of the words. To model the relationship between words and sentences, Single Value Decomposition(SVD) is used.The output of SVD helps in selecting sentences using the cross method. The sentences with the longest vector are selected.

Song et al., [23] suggested Fuzzy Evolutionary Optimization Modeling(FEOM) for clustering sentences. Consider 'n' objects that will be assigned to clusters according to the distances. The next step is to apply 3 evolutionary operators: selection, crossover and mutation till the termination condition is not reached(N_{max} = 200). There are 3 control parameters which regulate the crossover probability, p_c and mutation probability, p_m : Distribution Coefficient, Var, Relative Distance, G and Mean Evaluation effect, E_m. The best fitted sentences are then selected.

2) Supervised

Extractive Supervised summarization strategies diminish the weight of summarization by choosing subsets of sentences. Analysts working with NLP are particularly pulled in to extractive summarization. The fundamental focal point of the current work is to recognise remarkable highlights which would help in making a decision about the significance of a sentence in an article. A Supervised learning approach requires a lot of named information or labelled dataset. Extractive procedures select the top N sentences that best speak to the central issues of the article. Extractive procedures are set up as binary classification problems where the objective is to recognise the article sentences having a place in the summary. A directed methodology requires the presence of a bunch of reference, or gold, summaries. Accordingly, an administered model utilising a minimal robust set of features is the feature of the beneath strategies.

Collins et al., [25] considers the Recurrent Neural Network (RNN) framework. Since the sentences are randomly ordered in our dataset, there is no immediately available meaning for each sentence from the surrounding sentences. A set of such features are used for each phrase to provide the context locally and globally, which is described below.

1) AbstractROGUE

It is used for summarization as a feature. It uses the abstract, a pre-existing description, to manipulate the known structure of a paper. AbstractROUGE 's theory is that sentences that are strong qualitative summaries are often likely to have good summaries of the highlights.

2) Numeric Count

This is basically calculated on counting the times of numeric occurrences in a sentence, As sentences containing numbers/math do not contribute to a healthy summary.

3) Title Score

Non stop words in the text which match with those in the title are given more importance in the summary.

4) Keyphrase Score

Keywords used or predefined by the author are given more importance in the summary when used in the text.

5) Sentence Length

We add as an attribute the length of the phrase, an effort to catch the intuition that short phrases are quite unlikely to be successful summaries because they do not communicate as much data as longer phrases.

Charitha et al., [26] suggests that for an automated text summarization, Convolutional Neural Network (CNN) can be adapted by rating sentences. It learns features from sentences presented in a phrase and then allocates ranks to the same without needing any manual work by humans. As input, it takes word embeddings. Its ranks are produced as a part of the output for these phrases.'word2vec' module is used for the same.

Integer Linear Programming which is mostly known as ILP is generally used for sentence selection based on the ranks allocated previously by the model.

ILP attempts to solve problems where, keeping certain constraints, an objective function should be minimised or maximised, with the limitation that integers should be the variables used.

The CNN model takes the output from the above module word as an input. This CNN model is equipped to learn the characteristics of the sentences in order to rate them. A sentencing matrix which is the final matrix or input made by joining the vectors of the words of the sentence. Using word2vec, the pre-trained word vectors are created. There are multiple feature maps for each sentence in this model. Each feature map is created by applying filters to the sentence matrix.

For extractive summary, Wong et al.,, [27] examined combined sentence features. He used a supervised learning system to calculate the weights of various characteristics to determine how likely a sentence is to be meaningful.

A supervised learning classifier is then used after feature vectors of sentences are tested. In particular, candidate sentences are re-ranked, considering the duration of the final summaries is fixed. Lastly, to assemble the final summaries, the top sentences were removed.

Sentence Features(SF) is a collective term given by the author to show that certain words or phrases are considered much more weighted and important than other words/phrases in a sentence or a document based on their frequency, location and quote if any. Some of the other similar features considered, for Extractive Summarisation are:

1) Content Features

Based on content-bearing words, we combine three well-known sentence features , i.e., centroid words, signature terms, and high-frequency words, including representations of both Unigram and Bigram.

2) Event Features

An event consists of a term for an event and related event elements.

3) Relevance Features

To manipulate inter-sentence relationships, significance characteristics are integrated. Basic SVM needs the solution of the following optimization problem for a set of training examples. A hyper plane is used to distinguish research examples as positive and negative is predicted to be found here by the SVM classifier. The objective of Probabilistic SVMs is to estimate the

PERA & NG, 2010, [28] has developed a model using the Naive Bayes Classifier (NBC) by using a Na'ıve Bayes classifier to verify that going through the classifying text documents/phrases and using their summaries, instead of going through the entire documents, is really cost-effective. The Naïve Bayesian Classier insists that characteristics are autonomous. It learns the previous probability and the conditional probability of each function, and the highest posterior probability predicts the class mark.

B. Abstractive

1) Unsupervised

In Dohare, [2] the Semantic Abstractive Summarization (SAS) pipeline is developed. SAS first produces an Abstract meaning Representation (AMR) graph of the input story, in which it pulls out an abridged graph and finally, forms abridged sentences from this abridged graph. They developed a comprehensive approach to generating an AMR story graph using coreference resolution and Meta Nodes. They used an unattended novel algorithm depending on how people summarize a piece of text to extract a summary graph below. The pipe has three important steps. The 1st step is to convert the document to AMR. The 2nd step is to extract an AMR summary from the AMR document created in the previous step. The 3rd step is to create text from the extracted sub-graph. It surpasses previous methods of SAS technology by 1.7% and 3.7% using basic human solutions.

Chu & J. Liu [3] states that abstract summaries are made in data stocks of large, paired text documents. However, such data sets are rare and the models trained in them are rare in other domains. Recently, progress has been made in a sequence of sequences by single pairs. They only look at the places where there are only texts (product or business reviews) without the given summaries and suggest end-to-end constructions, which construct a neural model to create uncontrolled summaries. The MeanSum model has two key features:

1) A module of auto-encoder that learns the representation of each review and prevents the generated summary from being a language domain.
2) A module of summary that learns to generate meeting summaries similar to each input document.

The most common methods of using the abstract neural abstract are using supervised readings of many pairs to summarize the most expensive documents to find on the scale. The proposed model is not very catchy because it ignores attention or directions. The model does not offer a workable solution to a major problem (as there are a few guidelines for retrenchment) for a single document summarization problem.

Padmakumar & Saran, [4] suggests that group sentences shown at the top of the vector identify groups of similar sentences and select representatives for these groups to form a summary. The decoder is trained to specify embedding into sentences. They perform a combination of Sentence Embedding using RNN via Long Short Term Memory (LSTM), where they use a repetitive neural network with short-term memory to determine embedding into sentences. When representing sentences at a high vector level, the goal is usually to embed sentences directly or indirectly in such a way that the sentences closest to the definition are embedded next to each other in the vector space. Since sentences that form a group in a vector space may be close to each other, it is sufficient to keep one representative in each such group to make a summary.

Schumann [5] introduces an uncontrolled method of summarizing sentences in a logical way using the Variational AutoEncoder (VAE). It is known for its flexible mathematical learning, which represents high input. VAEs are trained in learning to reconstruct the input from potential variables. Providing explicit information about the length of discharge during training influences VAE not to include this information and can therefore be used during consideration. Instructing the decoder to produce a short output sequence leads to the output of the input sentence in a few words. The VAE system uses text data using RNNs such as encoder and decoder. The vectors μ and σ are formed from the last hidden coding state and the first codec cell state is started as z. They use a forward and backward bidding code. They show in different summed data sets, that these short sentences cannot form a simple foundation but produce higher ROUGE (Recall-Oriented Understudy for Gisting Evaluation) points than trying to construct a whole sentence. The idea that upgrading the decoder to produce shorter results will lead to more details expressed in a few words can be confirmed in the summary test. Linear Regression tests have shown that the length of the input sentence is inserted with the latest variable.

Zhang et al., [6] states that the method of summarizing the dialogue should take into account the context of many speakers where speakers have different roles, purposes, and language styles. In a tete-a-tete, as a consumer conversation, SuTaT aims to summarize each speaker by modeling customer words and agent words separately while maintaining their integration. SuTaT consists of a conditional production module and two unselected summary modules. The purpose is to generate a customer summary and a data agent summary. The design is similar to how a tete-a-tete is made: representative responses and customer requests depend on each other. Fine-looking models that can be rented perform better than unsupervised output models. Compared to other unstructured abstract foundations installed with LSTM encoders and decoders, SuTaT-LSTM has significant performance improvements.

Zheng et al. [7] states that the available summaries are based primarily on well-structured models in well-structured texts such as CNN and DailyMail news. Variations from the news, podcasts tend to be longer, more engaging and chattering, and more vocal about commercial and sponsored content, making the default Podcast Summarization a major challenge. They have designed two simple foundations for model comparisons: (1) Baseline 1: Choose the first tokens from the text as a summary. (2) Principle 2: Select the final tokens in the text as a summary. The idea of these two basic elements is that the end of a podcast can contain very important content details. Based on the basic analysis in this paper, we discuss many guidelines for future research: (1) Summary based on long narrative construction: Simple structure of heuristics is not the same as long narrative (2) Conversation summary: podcasts are conversational, interactive, and general. How to use existing research to help summarize podcasts is still in short supply. (3) Multi-module podcast analysis: We believe that multidisciplinary analysis is important in understanding the podcast and should therefore play an important role in summarizing the podcast and recommendations.

Yang et al., [8] is an uncontrolled abstract model with a denoising system that uses a transformer-decoder-based encoder-decoder structure and uses pre-training for massive unregistered power. The paper has three elements- (1) To use key sentences as the summary and to train the model to predict it during pre-training training. (2) Trained with the loss of the theme model and the denoising autoencoder. TED uses a multi-line decoder converter. (3) Instead of classical word tokenization, they use the SentencePiece tokenization. It follows the default configuration of the network converter.

Baziotis et al. [36] suggests a Sequence-to-Sequence-to-Sequence Autoencoder (SEQ3) sequence, consisting of two pairs of encoder-decoders. Here, words are used as a sequence of unintelligible variables. The first and last seq

Baziotis et al., [36] suggests a Sequence-to-Sequence-to-Sequence Autoencoder (SEQ3) sequence, consisting of two pairs of encoder-decoders. Here, words are used as a sequence of unintelligible variables. The first and last sequences are inserted and reconstructed sentences, respectively, while the middle sequence is a compressed sentence. The Embedding layer enables source sequence embedding coded projects by bidirectional RNN. To make a summary, we use the RNN decoder targeted, for their ground care and input feeds. The compiler works as a compressor, but its encoder works in the embedding of the abbreviated words.

2) Supervised

Abstractive supervised is a technique that creates summaries with words and sentences that are not present in the input text. It trains a supervised learning model with a dataset containing articles and their summaries. There are a few different supervised abstractive techniques that we have included in this paper.

Raphal et al., [10] gives a brief about various RNN variants used for Abstractive Text summarization. The basic RNN lacks in capturing the long term dependencies. This is rectified using Long Short Term Memory (LSTM) RNN model. LSTM consists of an input gate, output gate and a forget gate. It is used to capture long term dependencies. It also solves the Vanishing Gradient problem.

Khatri, Singh, and Parikh [11] implements the Abstractive Contextual RNN (AC-RNN) where a document context vector is passed as input at the first step to the encoder. The logic behind this approach is that if a person knows the context of a text, he/she can make the summary more easily as it provides a better understanding. This solves a major drawback by generating more document focused summaries rather than generic summaries.

Liu and Liu [12] presents a Supervised Abstractive Model using Conditional Random Fields (CRF) where the utterance compression is done as a sequence labeling task and is based on the Maximum Marginal Relevance (MMR). It uses BIO labeling scheme for sequence labeling. MMR score along with term weight of word is determined. The model selects the summary sentences, iteratively, until the given length limit is reached.

C. Reinforcement Learning

Reinforcement learning is used in text summarization to improve the efficiency of the existing techniques. This is done by training an agent with reward or punishment on every decision it makes and getting an optimal policy that would be used to generate summaries. In this paper we have discussed a few reinforcement learning techniques that are used for automatic text summarization.

Lee, & Lee [13] presents a Reinforcement learning model with Embedding Features. They have used the Deep Q-Networks (DQN)-based model. The sentences are represented as sentence embedding vectors. The Q-Values are computed using a Deep Neural Network model with a regression function. The Q-value is used to select the sentences. The role of agent is to select the sentences using sentence selector and generate a summary.

Prakash & Shukla [14] presents the Human Aided Text Summarizer "SAAR", for a single document setup. The input is passed through the preprocessing where the text is Tokenized and Isolated and a structured representation is created. The weights are calculated using ISF and IDF. RL is used to calculate the sentence score and a Term-Sentence Matrix is created. This is used to calculate the similarity using Euclidean distance and the sentence with the maximum distance is selected for the summary. The user checks the summary and if it is not adequate then the user gives feedback keywords according to which a new summary is generated.

Mohsen et al., [15] presents a Hierarchical Self-Attentive Neural Extractive Summarizer Via Reinforcement Learning (HSASRL) model. The first component is the Attention Sentence Encoder which has a bidirectional LSTM (Bi-LSTM) sentence encoder which encodes sentences into sequential representation vectors. Then the Attention Document Encoder composes a document representation. Then the Sentence Extractor labels the sentences as 1 or 0 according to their relevance in the summary. A learning Agent is trained to rank the sentences. It directly optimizes the ROUGE scores. The agent is initialized randomly and as it reads the documents it learns. The agent receives a reward for every match of its summary from the gold summary.

D. Hybrid

Hybrid approaches to text summarization are a combination of different techniques in order to counter their drawbacks. We have covered a few hybrid techniques in this paper.

Bhagchandani et al., [16] did research to build a Hybrid model that consists of three components - Clustering, Word Graphs, and Neural Networks. This model was developed for an abstractive multi-document setup. The model takes multiple documents and sends them to the preprocessing module where Normalization of passages is done. Files are tokenized to sentences and then to words. A single list of pre-processed strings is sent as input to the Summarization module where they are clustered and condensed to sentences. These are then ranked using TextRank which is an unsupervised extractive summarization technique A Seq2Seq Encoder Decoder model then performs sentence compression and the final summary is generated.

Wong et al., [27] has implemented a hybrid model for extractive text summarization through Probabilistic SVM and NBC. Supervised approaches to learning typically achieve good output but require data that is manually labelled. The amount of labelled data is decreased by co-training techniques. A co-training approach is developed to train different classifiers based on the same feature space. The combination of surface, material, and relevance features are added to PSVM and NBC. Co-training was applied to combine labelled and unlabeled data to decrease labelling costs. Experiments show that the semi-supervised learning method saves half of the cost of labelling and retains equivalent efficiency (0.366 vs. 0.396) compared with supervised learning. The ROUGE outcomes of the same summary process are enhanced.

PERA & NG, [28] implemented a new hybrid method which consists of 2 methods CorSum and Na'ıve Bayes classifier(NBC). The precomputed word-correlation factors are used by the CorSum architecture to classify representative phrases in a text to produce the description. In order to enhance the quality of CorSum produced summaries, CorSum-SF(CSSF) relies on word similarity.
The NBC is then used in large collections to classify CSSF created summaries of documents available on the web.
Rank values for CRSSF are identified on the basis of
1) The word-correlation(WC) variables that were previously added
2) Degrees of sentence similarity.
Talking about the WC factor and sentence similarity, it contains the non-stop, stemmed terms correlation variables, which is a 54,625 square symmetric matrix.
The correlation factor of each of the two terms wi and wj, which shows how closely wi and wj are related to semantics, is determined on the basis of the
1) Co-occurrence frequency.
2) Relative distance between Wi and Wj.

III. DATASET

TABLE 1. DATASETS FOR SUMMARIZATION

Dataset	Description
CNN/Daily Mail	It consists of both articles and summaries of long news articles.
Gigaword	It consists of nearly ten million documents, articles, and their headlines, (over four billion words) of the original English Gigaword Fifth Edition.
NYT	The New York Times dataset contains the full text and metadata of NYT articles from 1987 to 2007.
DUC	The Document Understanding Conference (DUC) archives and synopses assessed by the National Institute of Standards and Technology (NIST) since 2001.
20NG	It consists of 19,997 papers compiled in 20 separate categories from the Usenet newsgroup archive. 80 percent of the documents in 20NG were used for MNB preparation for assessment purposes and the remaining 20 percent for classification assessment.
TIDSUMM	TIDSUMM contains Darknet utilization information with 6831 documents of 26 distinct classifications crawled over the onion web or Tor network.
TTNews	A Chinese news summarization corpus, created for the shared summarization task at NLPCC 2017
SummMac	SummMac contains records about computer science gathered from ACL sponsored conferences.

IV. EVALUATION METRICS

The crucial step after generating a summary is to evaluate it. The summary can be evaluated by two methods: Automatic and by Humans. Automatic evaluation is a more feasible option than human evaluation because it is simple, fast and scalable. Text Summarization evaluation methods are: [24]

A. Extrinsic Evaluation

In this, the summary is checked on the basis of how the summary will help to accomplish other tasks, like information classification, answering questions, etc. successfully. For example, the reading comprehension is a summary about a given topic and helps to answer multiple questions. Therefore, a summary is good if it helps in completion of other tasks.

B. Intrinsic Evaluation

In this, the summary is analyzed between the automatic generated outline and human made outline. Intrinsic evaluation is done based on text quality, co-selection and content based.

The most prominent and frequently used intrinsic method of summary evaluation is ROUGE score, which comes under content based evaluation.

ROUGE(Recall Oriented Understudy of Gisting Evaluation) is an automatic summary evaluation method which calculates the score based on similarity between machine summary and human made summary. ROUGE score can be calculated by 5 ways: [37]

1) ROUGE-N: This measures the recall score based on a similar sequence of words in both the summaries called n-gram where n is the length of n-gram.

2) ROUGE-L: This gives a ratio between the size of Longest Common Subsequence between the two summaries and size of reference summary. It should be less than the unigram F score of both LCS.

3) ROUGE-W: This is the same as ROUGE-L only the weights, that is, common consecutive words.

4) ROUGE-S: It calculates the amount of skip bigram common between the two summarizes.

5) ROUGE-SU: This is an improvement ROUGE-S with weighted unigram.

V. CONCLUSION AND FUTURE SCOPE

In this paper, we have reviewed various research papers on abstractive, extractive, hybrid techniques along with learning methods - supervised, unsupervised and reinforcement. These papers have different algorithms and workings but all have promising results. Each of these techniques have their own set of challenges which can be solved using a certain variation in a particular technique. But the most significant and common challenges that are yet to be resolved are -

- Evaluation of the summaries: Judging the quality of the summary subjectively, both the manually written summaries and the summaries generated by the model.
- Labelled Data: To get more manually written summaries that can be fed to the model as training and evaluation data in supervised learning techniques.
- Anaphora Problem: To understand which pronoun used in the article is a substitution for which of the previously introduced terms.
- Cataphora Problem: To understand which ambiguous words or explanations are used to refer to a particular term before even introducing the term itself.

Future scope of Automatic Text Summarization is to resolve these challenges (and some other challenges) and make this technology more easier and feasible to implement. Research on Automatic Text Summarization is still going on to find the perfect model that can generate a summary like a real human.

REFERENCES

[1] Gonçalves, Luís. 2020. "Automatic Text Summarization with Machine Learning — An overview." Medium.com. https://medium.com/luisfredgs/au(Gonçalves,2020)Automatic-text-summarization-with-machine-learning-an-overview-68ded5717a25.

[2] Dohare, S., Gupta, V., & Karnick, H. (2018, July). Unsupervised semantic abstractive summarization. In *Proceedings of ACL 2018, Student Research Workshop* (pp. 74-83).

[3] Chu, E., & Liu, P. (2019, May). MeanSum: a neural model for unsupervised multi-document abstractive summarization. In *International Conference on Machine Learning* (pp. 1223-1232).

[4] Padmakumar, A., & Saran, A. (2016). *Unsupervised Text Summarization Using Sentence Embeddings* (pp. 1-9). Technical Report, University of Texas at Austin.

[5] Schumann, R. (2018). Unsupervised abstractive sentence summarization using length controlled variational autoencoder. *arXiv preprint arXiv:1809.05233*.

[6] Zhang, X., Zhang, R., Zaheer, M., & Ahmed, A. (2020). Unsupervised Abstractive Dialogue Summarization for Tete-a-Tetes. *arXiv preprint arXiv:2009.06851*.

[7] Zheng, C., Wang, H. J., Zhang, K., & Fan, L. (2020). A Baseline Analysis for Podcast Abstractive Summarization. *arXiv preprint arXiv:2008.10648*.

[8] Yang, Z., Zhu, C., Gmyr, R., Zeng, M., Huang, X., & Darve, E. (2020). TED: A Pretrained Unsupervised Summarization Model with Theme Modeling and Denoising. arXiv preprint arXiv:2001.00725.

[9] Wang, Y. S., & Lee, H. Y. (2018). Learning to encode text as human-readable summaries using generative adversarial networks. *arXiv preprint arXiv:1810.02851*.

[10] Raphal, Nithin, Hemanta Duwarah, and Philemon Daniel. n.d. "Survey on Abstractive Text Summarization." International Conference on Communication and Signal Processing, April 3-5, 2018, India.

[11] Khatri, C., Singh, G., & Parikh, N. (2018). Abstractive and extractive text summarization using document context vector and recurrent neural networks. *arXiv preprint arXiv:1807.08000*.

[12] Liu, F., & Liu, Y. (2013). Towards abstractive speech summarization: Exploring unsupervised and supervised approaches for spoken utterance compression. *IEEE Transactions on Audio, Speech, and Language Processing*, *21*(7), 1469-1480.

[13] Lee, G. H., & Lee, K. J. (2017, November). Automatic text summarization using reinforcement learning with embedding features. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)* (pp. 193-197).

[14] Prakash, C., & Shukla, A. (2014, September). Human Aided Text Summarizer" SAAR" Using Reinforcement Learning. In *2014 International Conference on Soft Computing and Machine Intelligence* (pp. 83-87). IEEE.

[15] Mohsen, F., Wang, J., & Al-Sabahi, K. (2020). A hierarchical self-attentive neural extractive summarizer via reinforcement learning (HSASRL). *Applied Intelligence*, 1-14.

[16] Bhagchandani, G., Bodra, D., Gangan, A., & Mulla, N. (2019, May). A Hybrid Solution To Abstractive Multi-Document Summarization Using Supervised and Unsupervised Learning. In *2019 International*

Conference on Intelligent Computing and Control Systems (ICCS) (pp. 566-570). IEEE.

[17] García-Hernández, R. A., Montiel, R., Ledeneva, Y., Rendón, E., Gelbukh, A., & Cruz, R. (2008, October). Text summarization by sentence extraction using unsupervised learning. In *Mexican International Conference on Artificial Intelligence* (pp. 133-143). Springer, Berlin, Heidelberg.

[18] Joshi, A., Fidalgo, E., Alegre, E., & Fernández-Robles, L. (2019). SummCoder: An unsupervised framework for extractive text summarization based on deep auto-encoders. *Expert Systems with Applications*, *129*, 200-215.

[19] El-Kassas, W. S., Salama, C. R., Rafea, A. A., & Mohamed, H. K. (2020). EdgeSumm: Graph-based framework for automatic text summarization. *Information Processing & Management*, *57*(6), 102264.

[20] Zheng, H., & Lapata, M. (2019). Sentence centrality revisited for unsupervised summarization. *arXiv preprint arXiv:1906.03508*.

[21] Vanetik, N., Litvak, M., Churkin, E., & Last, M. (2020). An unsupervised constrained optimization approach to compressive summarization. *Information Sciences*, *509*, 22-35.

[22] Ozsoy, M. G., Alpaslan, F. N., & Cicekli, I. (2011). Text summarization using latent semantic analysis. *Journal of Information Science*, *37*(4), 405-417.

[23] Song, W., Choi, L. C., Park, S. C., & Ding, X. F. (2011). Fuzzy evolutionary optimization modeling and its applications to unsupervised categorization and extractive summarization. *Expert Systems with Applications*, *38*(8), 9112-9121.

[24] Steinberger, J., & Ježek, K. (2012). Evaluation measures for text summarization. *Computing and Informatics*, *28*(2), 251-275.

[25] Collins, E., Augenstein, I., & Riedel, S. (2017). A supervised approach to extractive summarisation of scientific papers. *arXiv preprint arXiv:1706.03946*.

[26] Charitha, S., Chittaragi, N. B., & Koolagudi, S. G. (2018, August). Extractive document summarization using a supervised learning approach. In *2018 IEEE Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER)* (pp. 1-6). IEEE.

[27] Wong, K. F., Wu, M., & Li, W. (2008, August). Extractive summarization using supervised and semi-supervised learning. In *Proceedings of the 22nd international conference on computational linguistics (Coling 2008)* (pp. 985-992).

[28] Pera, M. S., & Ng, Y. K. (2010). A Naive Bayes classifier for web document summaries created by using word similarity and significant factors. *International Journal on Artificial Intelligence Tools*, *19*(04), 465-486.

[29] Bui, D. D. A., Del Fiol, G., Hurdle, J. F., & Jonnalagadda, S. (2016). Extractive text summarization system to aid data extraction from full text in systematic review development. Journal of biomedical informatics, 64, 265-272.

[30] Moratanch, N., & Chitrakala, S. (2017, January). A survey on extractive text summarization. In *2017 international conference on computer, communication and signal processing (ICCCSP)* (pp. 1-6). IEEE.

[31] Amini, M. R., & Gallinari, P. (2001, September). Automatic text summarization using unsupervised and semi-supervised learning. In *European Conference on Principles of Data Mining and Knowledge Discovery* (pp. 16-28). Springer, Berlin, Heidelberg.

[32] Krishnan, D., Bharathy, P., & Venugopalan, A. M. (2019, May). A Supervised Approach For Extractive Text Summarization Using Minimal Robust Features. In *2019 International Conference on Intelligent Computing and Control Systems (ICCS)* (pp. 521-527). IEEE.

[33] Shah, C., & Jivani, A. (2019). An Automatic Text Summarization on Naive Bayes Classifier Using Latent Semantic Analysis. In *Data, Engineering and Applications* (pp. 171-180). Springer, Singapore.

[34] Gillick, D., & Favre, B. (2009, June). A scalable global model for summarization. In Proceedings of the Workshop on Integer Linear Programming for Natural Language Processing (pp. 10-18).

[35] McDonald, R. (2007, April). A study of global inference algorithms in multi-document summarization. In European Conference on Information Retrieval (pp. 557-564). Springer, Berlin, Heidelberg.

[36] Baziotis, Christos, et al. "SEQ^ 3: Differentiable Sequence-to-Sequence-to-Sequence Autoencoder for Unsupervised Abstractive Sentence Compression." *arXiv preprint arXiv:1904.03651* (2019).

[37] Lin, Chin-Yew. "Rouge: A package for automatic evaluation of summaries." In Text summarization branches out, pp. 74-81. 2004.