IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 19 February 2024, accepted 6 March 2024, date of publication 13 March 2024, date of current version 21 March 2024.
Digital Object Identifier 10.1109/ACCESS.2024.3377463

RESEARCH ARTICLE

End to End Urdu Abstractive Text Summarization With Dataset and Improvement in Evaluation Metric

HASSAN RAZA AND WASEEM SHAHZAD
FAST School of Computing, National University of Computer and Emerging Sciences, Islamabad 44000, Pakistan
Corresponding author: Hassan Raza (hassan.raza@nu.edu.pk)

ABSTRACT Urdu, being a common language in South Asia, has not received significant attention in terms of language processing compared to more advanced languages. In the field of Natural Language Processing (NLP), the task of text summarization holds great importance due to its ability to comprehend textual content and generate concise summaries. Text summarization can be either extractive or abstractive in nature. While considerable efforts have been made to advance extractive summarization techniques, the limitations associated with it have been extensively explored and explained in the paper. However, the domain of abstractive summarization for the Urdu language remains largely unexplored. The challenges and underlying factors that have impeded progress in this domain have also been addressed. This paper specifically focuses on abstractive summarization of the Urdu language using supervised learning. To accomplish this, a labeled dataset consisting of Urdu text and its abstractive summaries is required. A dataset of Urdu text and its corresponding abstractive summaries has been prepared for the purpose of supervised learning. Additionally, the paper presents the results of summary generation, measured in terms of a rough score. Transformer's encoder-decoder network was employed to generate abstractive summaries in Urdu, yielding a ROUGE-1 score of 25.18 in Urdu text summarization. Moreover, a novel evaluation metric called the "disconnection rate" has been introduced as a context-aware evaluation metric to enhance the assessment of a summary, known as the Context Aware RoBERTa Score.

INDEX TERMS Datasets, neural networks, CA-RoBERTa score, text summarization.

I. INTRODUCTION

In this era of technology internet has strengthen every field. Due to internet the world has become global village for distributing the information. Every field distributes the data through internet and is stored for processing. Data is growing progressively over the servers on daily basis. It is also increasing in the form of text format like news papers, articles, magazines, blogs and e books [1]. Internet also provides a source of textual information like reviewers express their opinions on discussion forums in text form. Text summarization is the process of extracting the most salient and important information from a given text. It is the process in which text is condensed from larger to shorter one. It is the process of condensing a written content into a more compact form while retaining the most relevant information and meaning. Manual summarization is human resource hungry and time hungry process. Practically, it is very difficult for humans to manually summarize this huge amount of textual data [121].

In this busy world, life is too fast and everyone does not have so much time to read all raw data carefully [1]. To find the significant information from so much raw data, summarization of text is the need of hour. The more data is increasing, the more is the need of summarization. Automatic text summarization techniques are useful for quickly scanning a textual data [125].

There are two types of ATS, one is extractive text summarization(ETS) and the other is abstractive text summarization(ATS). In ETS, the most important and key sentences

The associate editor coordinating the review of this manuscript and approving it for publication was Juan Wang.

are identified and then combined to form the summary [47], [50], [126]. In ETS, the summary is extracted from the original text without changing the sentences [34]. Some algorithms and techniques are used to find the weight of each sentence in a document [25], [33], [54]. ETS is done by important sentence selection by a language modeling approach [45]. ETS techniques can be divided into three categories which are based on i) structural information or sentence position ii) unsupervised sentence ranking, and iii) supervised sentence classification.

The ETS approach is comparatively simpler than the ATS method, as it involves the application of specific algorithms to identify and extract only the most crucial sentences for summarization [18].

In ATS, the important text is paraphrased by natural language generation and understanding techniques. The complete text should be understood by the system and then sentence compression or reduction should be applied to find an abstractive summary. In this ever-growing research domain, extractive based summarized documents are grammatically correct and well-formed with a readability as compared to abstractive [18]. It is usually taken as 20% of the original length [18]. Summary can also be single document or multi-document. We are working in summarization of single document. Also summary is either Query oriented or generic. Our domain is to work in generic summary in which the summary will present the main theme of the document.

A lot of work has been done for the summarization of English text because there are no such ready resources of other languages. Urdu corpus and data sets are rarely available. Tag sets, word net and embedding vectors are also not so good as of English are. Urdu has a complex morphology and orthography.

The domain of abstractive Urdu text summarization has seen limited progress due to the numerous challenges it presents. There are also some limitations of ETS. First limitation can be considered when context between two consecutive sentences in summary is totally different. There is no any check for the context flow between them. The other limitation is if somebody have written a large article. Every sentence has minimum information coverage so we can't summarize maximum information in a summary of some sentences. Here abstractive summarization is needed to include information of many sentences into one.

From the literature, all the work done in the field of Urdu summarization is extractive. There is no any research article for abstractive summarization of the Urdu language articles. Due to not having the standard rules of writing Urdu and other factors described earlier makes Urdu difficult to interpret and learn by the computers. Today a lot of research is being done in this domain and there is research gap in this.

Although Urdu can't compete with the other languages which have standard rules of writing, parsing, interpreting and have ready resources to work in, but the presence of something, however small or imperfect, surpasses the absence of anything, for even the tiniest step forward carries more value than complete inaction. Some contributions have been added in pipeline of generating abstract summary of Urdu. We have addressed the three research questions. 1) How to prepare dataset for supervised learning of abstractive text summarization? 2) How to transform the transformer architecture in such a way that we can generate the summary of Urdu language and to built pipeline of embeddings and models? 3) How to enhance the evaluation mechanism of abstractive summarization? The main contributions of this paper are

- Prepared dataset for abstractive summary of Urdu documents.
- Generated abstractive summary with encoder decoder architecture of transformers and transfer leaning with addition of another layer of transformers for getting embeddings.
- Suggested Context Awared RoBERTa Score by adding disconnection rate for the improvement of summary evaluation

Contributions were made in three main areas of translation pipeline.

II. LITERATURE REVIEW

There is a variety of derivation and inflection in a single word that makes it challenging to work on language processing tasks [56]. There exist a number of variants in Urdu against a single word [122]. Urdu word segmentation is itself a problem because space is not the character to indicate a boundary. Urdu word segmentation has an important role in Urdu language processing. The challenges to work in Urdu language have been given in [56].

There are two types of text summarization, one is extractive text summarization and the other is abstractive text summarization. The majority of efforts have been dedicated to the advancement of extractive summarization techniques. It is also easy because we need to find the most important sentences of the document and combine those most n important sentences to form a summary. The primary aim of this extractive summarization is to find the importance of each sentence. Furthermore, it is worth noting that the sentence structure of the summary does not require specific attention, as it remains unaltered. The extractive summary preserves the original sentences in their entirety, including them "as is" without modification. Besides that, there are also some limitations of extractive summarization. First limitation can be considered when the contextual disparity observed between two consecutive sentences in the summary is entirely distinct. There is an absence of any mechanism or procedure in place to ensure the coherence and smooth transition of contextual flow between the consecutive sentences. Another limitation arises when dealing with extensive articles, as each sentence tends to possess a minimal scope of information coverage. Consequently, this hampers the ability to maximize information encapsulation within a summary composed of only a few sentences. In this context, the utilization of abstractive summarization becomes imperative as it enables the incorporation of information spanning across multiple sentences into a cohesive and concise representation.

In literature, diverse methodologies are employed for the purpose of summarization. Most of the work is done in the English language because it is an international and globally spoken language. It has simple morphology and proper delimitation in between words. There are predefined rules of grammar in English and, preprocessing is simpler than other languages. Hardly there is any work done in Urdu abstractive summarization, so techniques used for other languages have also been included in this literature. In this section first we have discussed approaches of English language. And due to lack of abstractive summarization approaches in Urdu, extractive summarization approaches and de-summarization approaches of Urdu language have also been discussed.

The methods of summarization can be divided into three categories i-e Machine Learning-based methods, Graph-based methods and Evolutionary algorithms based methods [18], [125]. In Graph-based methods, the graph of words is constructed and a summary is generated from that graph. Every method describes the way to find the weight of a sentence for the summary generation [47]. In evolutionary-based models, the summary of the text is usually extractive because the important sentences of text are found by evolutionary/genetic algorithms [51], [52], [54]. Niewiadomski et al. used fuzzy algorithms for the summarization of text. In [55], researchers introduced the concept of a type-2 linguistic summary of a database to compress the essential information from a large number of tuples containing both crisp and fuzzy values. The Zadeh fuzzy sets, which were generalized with type-2 fuzzy sets applied as models of linguistically expressed quantities and properties of objects, were used to generate natural language messages that can be used as supplements to statistical data analysis or as standalone textual information. Type-2 sets improve existing summarizing procedures by allowing linguistic concepts to be represented using a variety of membership functions found in databases. In the same way that a type-2 fuzzy set widens the idea of a standard fuzzy set, the method expanded type-1-based methods, with the former encompassing the latter as a particular case. The type-2 forms of summarizers, quantifiers, and searches have been specified, presented, and implemented in summarizing databases. This work is done in summarization but this is summarization of database. Jing also used sentence reduction algorithm in which they describe a one-of-a-kind sentence reduction approach that eliminates extraneous phrases from sentences obtained from a document for summarizing [59]. The algorithm assessed a range of sources of information, including syntactic knowledge, context information, and statistics generated from a corpus of samples published by human experts, to identify which phrases in an extracted sentence may be eliminated. They devised a sentence reduction strategy that is fully automated. The source content, as well as the extracted phrases, are fed into the reduction procedure. As reduction outputs, reduced versions of the extracted sentences are created, which may be used to build summaries directly or in combination with other sentences. The purpose of this study was to determine which phrases in a sentence are less critical and might be removed for a specific sentence in a particular context. They also created a corpus of 500 phrases and their condensed variants produced by human experts, which they utilized to train and test the system. This work is focusing on extractive summarization.

Merchant et al. and Sun et al. used reinforcement learning for the summarization of text. In [32], reinforcement learning has also been used for an Indian judiciary data but most of the techniques are supervised learning techniques in literature. Reason for using reinforcement learning instead of supervised learning for judiciary data is due to uniqueness of pattern in each case. Also because criminal and civil judgments have different patterns. Semantics of text was also included while summarizing the articles. Merchant et al. employ latent semantic analysis, which evaluates similar concepts inside a piece of text, in the proposed latent semantic analysis system. Another difficulty was targeted because legal statistics is that criminal and civil judgments have different patterns. While each civil case is unique, criminal proceedings tend to follow a similar pattern. As a result, they employ two methods: a single document approach and a multi-document strategy. Their data is made up of legal decisions made by the Indian judiciary. The key concepts from the original paper were preserved in a brief synopsis. They were able to get a ROGUE-1 score of 0.58 on average [32]. Sun et al. used the latent semantic analysis for summarizing the legal documents. They don't use deep neural networks / machine learning methods because every case has its context and the deep neural networks decide based on previous data. Sun et al. used reinforcement learning for text summarization. They produced summaries through reinforcement ranking on semantic link network. They have claimed that semantics of text is not considered while summarizing the text so they ranked sentences on semantic link network. A method was suggested in the literature to extract the association semantic rules and use those rules for the summary generation [25]. As the judicial case have multiple hearings and multiple documents. So the last two papers focused on multiple documents summary but our work is for the single document.

Machine Learning-based models are also used for text summarization tasks [18], [19], [31], [33], [41], [45], [57], [124]. In these types of methods, the Recurrent neural networks, sequence to sequence, and reinforcement models are used. Both extractive and abstractive summaries are extracted by ML-based methods. But in abstractive summary, the GA based and Graph-based methods are used rarely. For abstractive summarization, usually the machine learning, deep neural network and reinforcement models are used. For abstractive summarization supervised as well as unsupervised approaches are present in literature. Following [45] and [41] have used the deep neural networks for supervised learning of summarization. In [45], extractive summarization has been done by recurrent neural network language modeling for English language. Researchers proposed a framework based on recurrent neural network language modeling (RNNLM), which can render not only word usage cues but also long-span structural information of word co-occurrence relationships within spoken documents, bypassing most existing LM-based methods' strict bag-of-words assumption. Additional in-depth descriptions of modeling features and corresponding summary performance of various instantiation techniques and a study of alternative model complexity and combination tactics are included. In addition, comprehensive comparisons with several state-of-the-art unsupervised summarizing algorithms demonstrate the effectiveness of our RNNLM-based techniques. This technique is unsupervised and their results are not significant as compared to supervised methods. While Nallapati et al. proposed neural encoder-decoder and attention models for abstractive text summarization [41]. As input, they employ words, morphemes, and embeddings. Bidirectional RNNs and LSTMs are employed. Their work builds on the framework established but adds new models that solve fundamental issues in abstractive summarization. The fundamental model is the encoder-decoder model with attention and a broad vocabulary trick. A feature-rich encoder is used to handle the difficulty of capturing essential concepts in a topic. A switching generator/pointer system is modeled to cope with Out-of-vocabulary words. In the case of extensive texts, significant sentences must be recorded in addition to keywords. Hierarchical attention is used to accomplish this. English data sets were used for this architecture. According to See et al., existing abstractive summarization algorithms, according to experts, have two flaws: they present factual facts incorrectly, and they occasionally create repetitious output [109]. An abstractive summarization architecture is proposed in this paper, which combines a hybrid pointer-generator and coverage with the classic attention-based seq2seq approach. The hybrid pointer-generator may copy words from the input text by pointing, resulting in accurate information replication while allowing the generator to produce new words. Although this gives precise data, it is still repetitious. Using a coverage method that retains a record of material that has already been summarized eliminates repetition. This work is also done for the English language. And transformers enhanced the results of this architecture. In [110], Harvard researchers extended the [109] by adding the attention techniques in bottom-up way. The content selector here predetermines the terms included in the summary and then uses this content to limit the neural model to generate an abstractive summary. This model performs better on the CNN/Daily Mail and New York Times corpora. Another abstractive summarization technique [19] was presented by the Google researchers based on transformer based architecture, PEGASUS (Pre-training with Extracted Gap-sentences Abstractive Summarizing Sequence to Sequence Models). We extended this architecture to use for Urdu language. The encoder-decoder design is based on a typical transformer-based encoder-decoder architecture. For numerous abstractive summarization tasks, the authors first established a pre-training aim. This self-supervised aim was dubbed GSG (Gap Sentences Generation). Their approach was identified as fundamental sentence selection. C4 (Colossal and Cleaned version of Common Crawl) and HugeNews datasets were utilized for pre-training. ROUGE scores were used to evaluate the proposed approach. Another work was done by Zhu et al. using the transformer architectute. Microsoft's speech and dialogue research department created an abstractive text summarizing system for automatically generated meeting transcripts [111]. Zhu et al. contend that meeting summarization varies from document summarizing. Because a meeting involves many participants, meeting transcripts take many different forms due to differences in semantic styles, ideas, and roles. The proposed meeting summarizing architecture, the Hierarchical Meeting Summarization Network, is based on deep learning technologies. The HMNet model uses the encoder-decoder transformer architecture to provide abstractive summaries of meeting transcripts. On the AMI and ICSI meeting corpus, the model was evaluated using three variations of the ROUGE measure. But in this paper there are multiple transcripts, one of each speaker. So the dynamics of summarization vary with this environment. In supervised machine learning category, a multilingual model was also proposed in Multi-Summ [112]. It is an abstractive summarizing model capable of handling a variety of languages, including English, Chinese, French, Spanish, German, Bosnian, and Croatian. This is also supervised leaning method. Multilingual training (language model, auto-encoder model, translation, and back translation model) and joint summary generation training were implemented in two steps by the researchers. In addition, a new summary dataset for Bosnian and Croatian languages is being developed. The Transformer architecture is used to implement the model. The Europarlv5 dataset was utilized for English, German, Spanish, and French, while the News-Commentary-v13 dataset was used for Chinese, and the SETIMES dataset was used for Bosnian and Croatian. This paper has not targeted the Urdu language due to low resources and complex morphology. Then Kryscinski et al. identified two significant improvements to summarizing models that increase the amount of abstraction in the summary while maintaining word overlap with the ground-truth summary. To separate the extraction and creation operations of the decoder, they employed a contextual network and a language model. A hybrid objective was also adopted, which increased n-gram overlap with the ground-truth summary while encouraging abstraction. In this approach different two models were used which would cost more than a single model and technique is evaluated on English. Encoder and decoder networks are used for summarization but Li et al. modified the encoder-decoder architecture by adding an information selection layer to represent the information selection process, inspired by human summarizers who first read the manuscript and deleted extraneous items. The information selection layer is a gated global information filtering network with two parts: gated global information filtering for removing extraneous data and local sentence selection for progressively selecting salient phrases from a document to provide a summary. This approach has also been applied to the English text. Perez-Beltrachini et al. introduced deep communicating agents in an encoder-decoder architecture to solve the difficulty of expressing a lengthy text [115]. Each agent uses two stacked encoders to encode a paragraph: a local encoder fed into the contextual encoder, and a contextual encoder broadcasts their encoding to others. The agents will communicate global context information about different portions of the document in this fashion. The agents can cooperate and focus on the relevant portions of the input text by forwarding new messages via different levels. The decoder is an LSTM with additive attention applied to the agents to integrate information. Cohan et al. further modified encoder-decoder as hierarchical encoder and discourse-aware decoder. The abstracts of scientific articles were employed as ground-truth summaries since they were significant texts with discourse information [116]. A hierarchical encoder captures the document's discourse structure, and a discourse-aware decoder creates the summary in the suggested model. To capture the discourse structure, the word-level BiLSTM converts a section's word sequence into vector representations, which are then fed into the section-level BiLSTM to generate the document representation. Discourse-aware attention was proposed, which uses discourse-related information to adjust word-level attention. A copying method and a coverage model were also used to cope with OOV words and self-repeat difficulties. The authors browsed over 340 000 lengthy and structured scientific publications from arXiv.org and PubMed.com as their experimental dataset. After discussion of abstractive and extractive methods, Chen et al. utilized reinforcement learning to combine extractive and abstractive summarizing, based on how individuals summarize vast amounts of information [117]. The most crucial sentences are picked first, and the rest of the phrases are rebuilt. In the sentence extraction network, the temporal convolutional model is employed to build the representation of each phrase. A BiLSTM is used to include the document's whole context. A second LSTM trains a pointer network to retrieve phrases regularly. OOV phrases from the input material are copied directly into the output summary using the copying method. This [1] is almost the only paper in which summarization is done in Urdu Language but the summarization is extractive. The sentence weight algorithm was used to extend the single document extractive desummarization approach for Urdu, specifically for news, sports, and health issues. Bhatti et al. preprocess the material and use a sentence weight algorithm to encapsulate it. The input text was divided into sentences. The tokenization approach was used after segmentation. The suggested tokenization approach separated each word from the phrase. They gathered all the stop words from the input text after tokenization. Their suggested method examined each word, except stop words, and then applied de-summarization after locating the contextual word [1]. Most relevant approaches and papers have been given in the table of literature review for comparison.

III. DATASET PREPARATION FOR ABSTRACTIVE SUMMARIZATION

To get an abstractive summary from a model, we need to train a model on same type of data. Also deep learning techniques need a huge amount of labelled data to best generalize the concept. For making a model learn to generate abstractive summarization, we need to provide the Urdu text documents along with their respective abstractive summaries. And for attaining very goods at test time, we should train model with such type of data which holds all the possible scenario but it is impossible to generate such type of data, because language has no finite set of sentences and their summaries. Maximum effort can be put and in one go, the data can not be released will all the specifications. Generation of dataset is also a research task in which researches participate to enhance the dataset. There was only one dataset of Urdu summary consisting of only 50 documents [40].

Collection of dataset is also a resource hungry process. There were two ways to make dataset, one is to scrap the Urdu documents from web and collect articles from any source. Then we would have to hire domain experts of Urdu and ask them to write summaries. Experts would also provide summaries in hard form written on the papers through pen. Then we had to utilize human resources to type in computer to get summaries in soft form. This process was very lengthy and was to start from zero for writing of summaries and we would not be able to prepare a significant amount of data for this problem. The other way was easy and same amount of resources produced more amount of data.

Deep learning techniques a huge amount of data to best model the problem, which is going to be solved. Because if we don't have the enough amount of data to make the model learn the distribution or we don't have the data for all possible scenarios, We can't model the problem. Due to small size of data, models can over-fit to learn and it can't be applied to real world problems. We need the dataset which covers maximum variations of sentence structure. There is only one dataset for the summarization of Urdu text [40] which contains only 50 documents of Urdu with the summaries. We prepared dataset of 19615 documents with the summaries. We took an open source dataset of English and prepared the same data for Urdu language. A step wise translation was made. We translated documents and summaries from google. Translation from the google was done because it is easy to correct the only wrong translated sentences as compared to write the new translations. Figure 1 explains this step of translation.

Typing of Urdu is also a human resource hungry process. Also the keyboards are mostly available in English. Typing of English is much easier than in Urdu because the alphabets

TABLE 1. Literature review.

Reference	Language	Methodology	Limitations
[45]	English	RNN Language Model	This technique is unsupervised and their results are not significant as compared to supervised methods.
[41]	English	encoder-decoder model with attention	occasionally create repetitious output
[124]	English	hybrid pointer-generator and coverage with the classic attention-based seq2seq approach.	attention was added in top down way which caused limitations
[110]	English	attention techniques in bottom-up way. Content selector here predetermines the terms included in the summary and then uses this content to limit the neural model to generate an abstractive summary.	Main focus was the selection of the content which is more likely to be extractive summary which selects important sentences.
[19]	English	Transformer based architecture. We extended this architecture for summary generation. Evaluated thorugh ROUGE Score	Only english was targeted and ROUGE was used as evaluation metric
[111]	English	The HMNet model uses the encoder-decoder transformer. summarizing system for automatically generated meeting transcripts in which there are multiple speakers	But in this paper there are multiple transcripts, one of each speaker. So the dynamics of summarization vary with this environment.
[112]	Many	Multilingual training on tarnsformer based architecture The Europarlv5 dataset was utilized for English, German, Spanish, and French, while the News-Commentary-v13 dataset was used for Chinese, and the SETIMES dataset was used for Bosnian and Croatian.	This paper has not targeted the Urdu language due to low resources and complex morphology.
[1]	**Urdu**	Extractive Summarization by using sentence weight algorithm	Our approach can't be compared with this due to extractive summarization

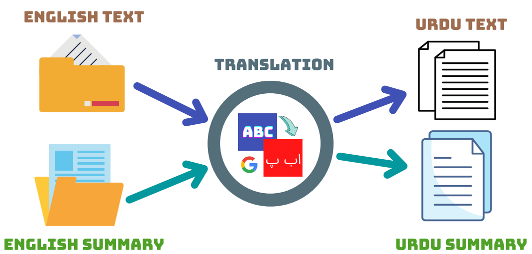

FIGURE 1. Dataset preparation process.

FIGURE 2. Dataset preparation process.

of English are 26 and or Urdu are 36. The combinations of alphabets increase exponentially as the number of alphabets increase. Also Urdu has complex morphology. There are no uniform rules for writing Urdu words as English and other languages have. In Urdu it is not necessary to delimit each word with the space. There are some alphabets of Urdu which do not join with other words وطن is a single word in which three alphabets are combined و,ط and ن.. From these three alphabets the first one is non joiner. Other joiner alphabets can join this as last alphabet but it can not further join others. In English each alphabets seems to be different and words are always delimited by space while in Urdu it is not always true.

The translation of google is not fully accurate, so we had to manually made those translations correct. Manually correction of documents was a human resource intensive task and it was also necessary to produce error free and valid summaries.

First we used google API to translate the sentences. We parsed English documents to get sentences. Raw data of English was hard to parse because it was mostly unstructured. Each sentence was translated separately. We parsed the documents sentence by sentence and get it translated from the google and then translation was concatenated to make the Urdu document. Same steps were applied for the translation of summaries. We made separate columns for Urdu text, English text, Urdu summary and English summary. Each row contained a record. These data rows were distributed among the language experts to correct the wrong translations. The second step of manually correcting the sentence has been shown in the Fig 2. The data received from the experts was combined for passing it to the deep learning model.

Figure 3 is the comparison of datasets already available and ours. We had to use log-scale for comparison of numbers. Without using log-scale, the bar of existing dataset was invisible.

In the domain of deep learning, data is just crud oil and the machine learning techniques are refineries to extract useful things out of data.

This is sequence to sequence problem in which sequence of text document is given along with the summary during the training of model while at test time, only text documents is given and the sequence of word of summary is predicted.

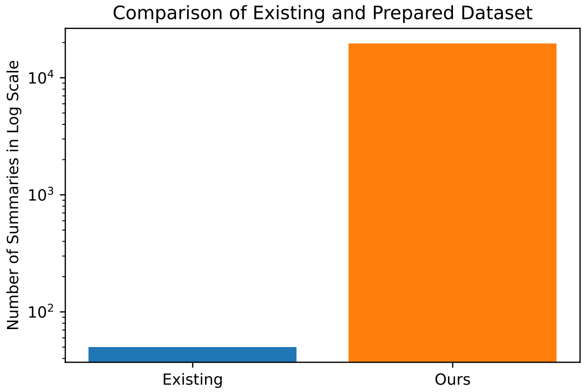

FIGURE 3. Dataset comparison.

IV. METHODOLOGY

In NLP tasks, deep learning has become the important part and Transformers have shown tremendous improvements in the results. Transformers are considered to be the state of art methods to solve any problem in the domain of NLP. The advancement in Deep Learning mainly depends on the pretraining and architecture of the model [63]. Transformers overcome the first dependency of deep learning methods and provide the facility of Transfer learning.

According to the core concept of transformers, transformer models are beings open source because there are thousands of models available in [https://huggingface.co/models] Transformer Models Repository. For every task related to transformer architecture, we can download model and place it in the project to further fine-tune the parameters by using specific dataset of a particular problem. After training that particular model you can upload that trained model to the public repository. In this way, every individual can contribute to this research community. Easy way of transformer has been discussed and it can be used for the similar tasks or for the similar languages but the problems and aspects have already been discussed, where Urdu varies the other languages. We can't directly use Urdu to train the transformer model. Both encoder and decoder receive embeddings of sources and targets respectively. Any type of embedding(TF-IDF, Positional vector, count vectorizer or any other) can be given. Transformers have the tendency to learn embeddings and contextual information. So both the non-contextual and contextual embeddings are used as inputs of sources and targets. A detailed study was conducted and did different experimentation, to find the best embedding of Urdu language. Those embeddings or vectors were used as input in transformer architecture.

At the start, one hot encoded vectors were given as input to any model. Dimension of those vectors were equal to the size of dictionary of that particular language. There were two main shortcomings of those vectors. First those vectors were very sparse and all the entries of vector was zero except one entry. Zeros don't convey any useful information. Other shortcoming was lack of context in that vector. Possibly there can be a vector which has non zero values along

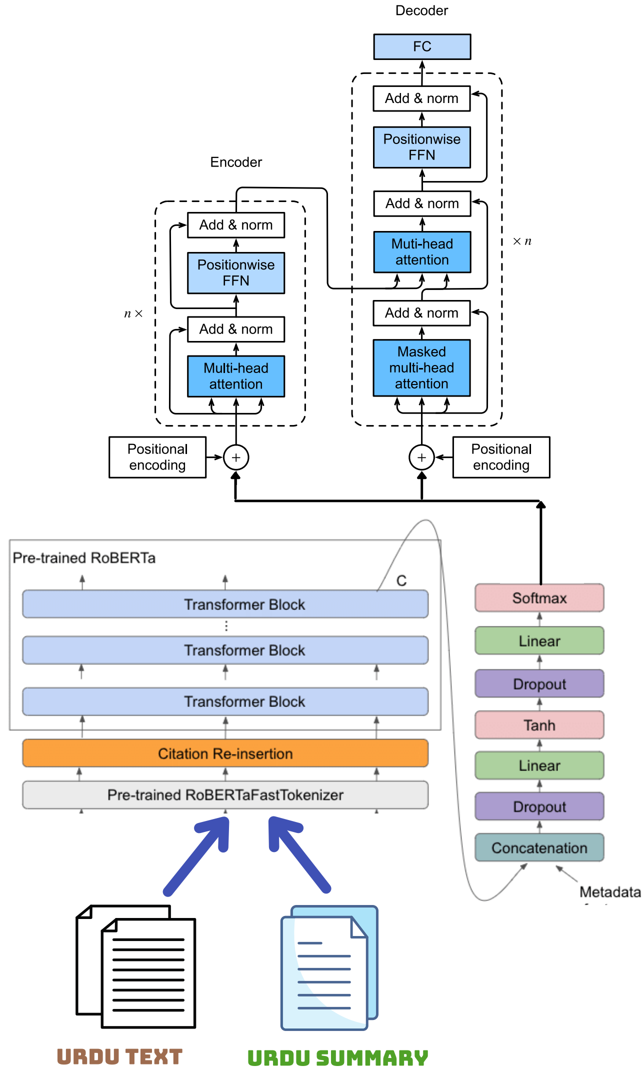

FIGURE 4. Proposed methodology for abstractive summarization.

the dimensions and those value can preserve the context. Besides these, the length of vector can also be reduced by preserving the context and replacing the zeros with useful numbers. First solution to this problem was solved by word to vector [118]. In word2vec architecture encoder and decoder networks were trained combined. Encoder condenses the vectors and decoder reconstructs the surrounding words given center word or vice versa. Word2vec can either be Continuous Bag of Words(CBOW) or Skipgram depending upon whether surrounding words are predicted given center words or vice versa. Google had released word2vecs of English vocabulary. Word2Vec is one of the best embedding technique. After Word2vec LSTM and RNN based encoding techniques were suggested as BERT [104]. Bidirectional Encoder Representations from Transformers (BERT) was enhanced by RoBERTa [119]. RoBerta is state of the art embedding technique.

Instead of passing simple embeddings (one hot encoding or positional encoding) to encoder and decoder networks of

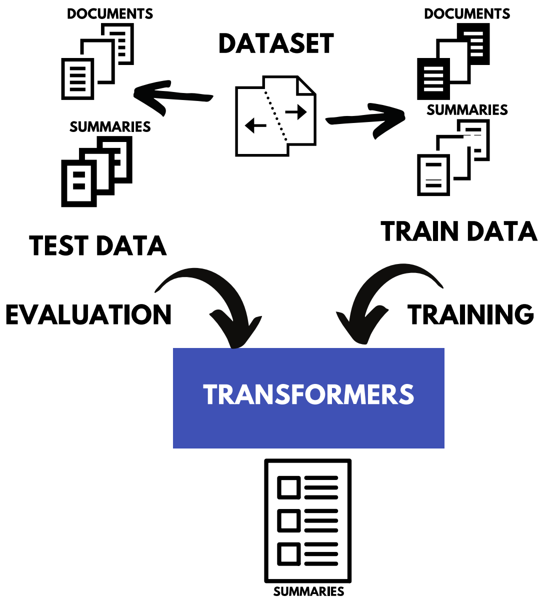

FIGURE 5. Process of training and evaluation.

Transformer architecture, we can pass a very efficient and effective RoBERTa embeddings. RoBERTa itself contains the transformer architecture. So overall two transformers are required to generate summary. One is to generate the contextual embedding and other is to generate summary. Overall methodology has been represented in Fig 4. Along with the other two main contributions of dataset and CA RoBERTa Score, architecture of abstractive summary has been presented with a variety of results. The general issue of stacking different layers and making a very deep network is of vanishing gradient. Problem is either expressed as the vanishing gradient or problem with the flow of gradients in back propagation of neural network. The network stops learning due to saturation of gradient but here both the networks are not trained combined. The network for the RoBERTa embeddings is trained separately and the other of summary generator is trained separately. Both have their own objective functions. For the network of embeddings attention is defined as:

$$Attention(Q, K, V) = softmax(\frac{Q.K^T}{\sqrt{t}})V$$

There is a significant difference between the results of using single transformer architecture and two layers of transformers.

Overall a big picture of the training and evaluation process has been shown in Figure 5 and 6.

We propose a novel approach which extends the architecture of Transformers Encoder Decoder. Some extra features form the given documents are stacked at the initial layers of encoder network to give more attention. Stop words don't have direct impact on summary which a model should learn.

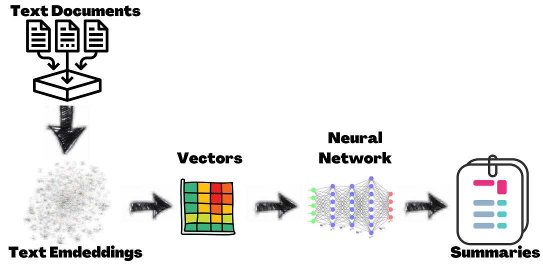

FIGURE 6. Process of training and evaluation.

V. DISCONNECTED RATE AS A FACTOR OF EVALUATION METRIC

The tasks in which some text is generated by the system are evaluated by comparing that generated text with a reference text. Either the text generation is supervised or unsupervised, system generated text is compared with ground truth labels. In most of the evaluation metrics, n-grams are matched to report the correctness of a model. For n grams both the reference and candidate text is tokenized into unigrams. Given a reference text x, its i tokens represented as $x = \langle x_1, x_2, x_3, x_4, x_5 \ldots .x_i \rangle$ and a computer generated text as $\bar{x}$ and its j token represented as $\bar{x} = \langle \widehat{x_1}, \widehat{x_2}, \widehat{x_3}, \widehat{x_4}, \widehat{x_5} \ldots .\widehat{x_j} \rangle$. An evaluation metric will be some function $f(x, \hat{x}) \in \mathbb{R}$. There must be higher correlation between human judgement and the value of this function.

Let S_x^n and $S_{\hat{x}}^n$ be list of tokens of n-grams($n \in \mathbb{R}_+$) in reference text x and candidate text $\hat{x}$ respectively. The number of matched n-grams will be $\sum_1^n(w)\forall w \Leftrightarrow (x == \hat{x})$. For all those n-grams which are equal in both texts is summed up for evaluation. If exact matching is not being done or some transformations are also done for matching we can write as

$$\sum\nolimits_{w \in S_x} \prod [w \in S_{\hat{x}}] \tag{1}$$

where $\prod$ is any transformation function like synonyms or stem word. Precision and Recall of matching n-grams is given below

$$Precision{-}P_n = \frac{\sum_{w \in S_{\hat{x}}} \prod [w \in S_x]}{|S_{\hat{x}}^n|} \tag{2}$$

$$Recall{-}R_n = \frac{\sum_{w \in S_{\hat{x}}} \prod [w \in S_x]}{|S_x^n|} \tag{3}$$

Most of the metrics depends upon these scores. These are the evaluation metrics to evaluate the computer generated text [64], [67], [68], [69], [70], [71], [72], [73], [74], [75], [76], [77], [78], [79], [80], [81], [82], [83], [84], [85], [86], [87], [88], [89], [90], [91], [92], [93], [94], [95], [96], [97], [98], [99], [101], [102], [106]. BLEU score [67] is popular evaluation for text generation or translations models which suggests some modifications in Precision. It is also n-gram matching evaluation metric. Reference [68] suggested to compute BLEU score on sentence level SENT-BLEU. It does not take accountability of the context of a sentence.

METEOR [106] computes P_1 and R_1 but during computations it does not only matches the exact words but also consider the synonyms, word stems and paraphrases as matched one. METEOR 1.5 [43] assign weights to all the factors of evaluations. METEOR++ 2.0 [70] incorporates external paraphrase resource. NIST [71] is modified form of BLEU score that assign weight to each n-gram. δBLEU [72] modifies multi-reference BLEU by including human annotated negative reference sentences. CHRF [73] compares charcters n-gram and on the basis of that, it suggests an evaluation metric but same context is being ignored which is the main contribution. CHRF++ [74] extends the previous work by adding bi-grams.

Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [64] Score is used for evaluating the summary generated by the system. Its different variants are used for the comparison of results. Most of the researchers used this evaluation metric for comparison. ROUGE-1, ROUGE-2 and ROUGE-L are mostly used. These all are the variations of ROUGE depending upon the number of words which are being matched. ROUGE-1 compares the uni-grams and ROUGE-2 compares Bi-grams. ROUGE-L take the longest common sub-sequence.

Some methods use the word error rate or word edit distance from the reference text [78]. It quantifies the evaluation by taking account of edit operations required to make reference text from the candidate text. TER [79] normalizes the edit distance by $|S_x|$. ITER [80] also consider the stem words as matching one, and suggests better normalization. CHARACTER and EED calculates the edit distance based on characters and achieve higher level of correlation with human judges for some languages [83], [84].

Word embeddings are the dense vector representation of a word. While encoding a textual data or categorical data, spare vectors are formed thorough any embedding technique but there are two problems in sparse embedding of any word or sentence. It consumes a very large amount of data as compared to dense representation, either one or two values are ones while all other values of vector are zeros. Secondly it contains a very limited context. So work is continuously being done in this domain to represent such data in numerical form that it contains the context and is not sparse. Word2Vec (Word to Vector) [107] architecture was made to make context awared embeddings. Encoder and decoder network was trained as a single network to make context awared embeddings. Given a sparse vector the encoder gives a dense representation and given that intermediate representation to decoder network, decoder was to transform back to sparse vector as given in encoder by preserving context (neighbour/surrounding words). Further word2vec can either be Skip gram or CBOW (continuous bag of words) depending on whether given surrounding words middle is predicted or vice versa.

Different architectures of word embeddings are given in [85], [86], [87], [88], and [89]. MEANT 2.0 [90] uses word embeddings and shallow semantics to structural and lexical similarity. YISI -1 [91] is extension of MEANT2.0 by making optional the context parses. The matching can be either greedy or optimal based on the number of words being compared. There should be a trade off between greedy and optimal matching which has been studied and presented by [97] and [98]. From the literature [19], it has been proved that context based metrics are better evaluating the summary or any candidate text given a reference text.

Abstractive summarization is also evaluated with the same evaluation metrics. Research is also being done to improve the method of evaluation. But logically the abstractive summarization should not be evaluated by this metric because summary can have the same context but not the exact same words of reference summary. In this case the ROUGE score will not give good evaluation to the summary.

Many researchers are trying to improve the evaluation metric for summarization task [9], [11], [19], [26], [43]. ROUGE AR was suggested in literature in which anaphora resolution was done and edit distance was made as a factor of evaluation metric. Latent semantic analysis and singular value decomposition were used to cater with the textual redundancy and coherence [65]. After the ROUGE-AR, ROUGE 2.0 was proposed in which synonyms of the words were also considered as a count of correct words generated [66]. In ROUGE 2.0 researchers reported the results by making the variation of removing the stop words and matching the synonyms. But still a good score can be generated even if the context of the sentence is completely the opposite.

In [11] a new evaluation metric as BERT score was suggested in which system summary was related with reference summary through a Deep Neural Network. Contextually sentences of both summaries were compared and reported the correctness of summary based on the score. BERT architecture was used to computer context vectors of sentences. But still there are some factors on which summary is not evaluated. Problem which is being discussed can have more frequency in extractive summary but also exist in abstractive summary. If there is a sentence at the start of documents and other is from the last of documents which are out of context from each other and both have been selected for the summary. The quality of summary will drastically decrease due to this problem and it will become a bad summary in human judgement. So this factor should also be an important factor while evaluating the summary. From the contextual measure given in [11] and the problem we discussed we can derive two concepts for evaluation.

Not only strong correlation between the system generated summary and reference summary should be focused but also we must focus that how much is the diversity of two consecutive sentences in a summary. The similarity of contextual vectors of both the summaries should be rewarded positive while the difference of contextual vectors(disconnection rate) within the summary should be reinforced negatively. Evaluation score is directly proportional to the similarity of contextual vectors of reference summary and system

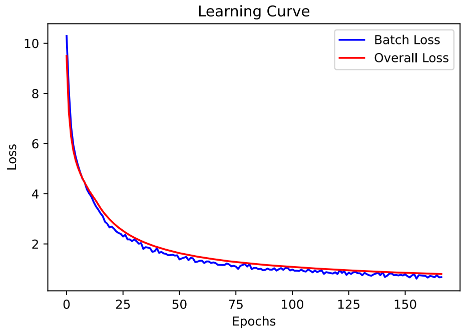

FIGURE 7. Learning curve loss vs epochs.

summary. Also evaluation score has inverse relation with the disconnection rate within the summary. We name this evaluation metric as Contextual Awared RoBERTa Score. Given a reference text x, its n tokens represented as $x = \langle x_1, x_2, x_3, x_4, x_5 \ldots . x_n \rangle$ and a computer generated text as $\bar{x}$ and its m token represented as $\bar{x} = \langle \widehat{x_1}, \widehat{x_2}, \widehat{x_3}, \widehat{x_4}, \widehat{x_5} \ldots . \widehat{x_m} \rangle$, CA RoBERTa Score can be expressed as:

$$CARobERTaScore \propto similarity(x, \hat{x})$$

Given computer generated summary as $\bar{x}$ and its o tokenized sentences represented as $\bar{x} = \langle \widehat{s_1}, \widehat{s_2}, \widehat{s_3}, \widehat{s_4}, \widehat{s_5} \ldots . \widehat{s_o} \rangle$

$$CARobERTaScore \propto -DR(\hat{s}_i, \widehat{s_{i+1}})$$

where similarity function has been defined below and disconnection rate(DR) is the measure of disconnection between context of consecutive sentences within a system summary.

$$DR(\hat{x}) = \sum_{j=1}^{o-1} \frac{d(\hat{s}_j, \hat{s}_{j+1})}{o-1} \quad (4)$$

In above equation d is Euclidean Distance between consecutive sentences of system summary. Let each sentence embedding is $\hat{s} \implies V[t]$ t dimensional vector.

$$d\left(\hat{s}_p, \hat{s}_{p+1}\right) = \sqrt{\sum_{i=1}^{t} \left(\hat{s}_i^p - \hat{s}_i^{p+1}\right)^2}$$

where p is the position of sentences in system summary. Its value will start from zero to m-1 as given in upper equations.

$$similarity(x, \hat{x}) = \sum_{j=1}^{m} \sum_{i=1}^{n} \frac{cos(embd(x_i), embd(\hat{x}_j))}{m+n} \quad (5)$$

and

$$COS(x, \hat{x}) = \frac{x.\hat{x}}{|x|\,|\hat{x}|}$$

For finding the similarity of x and $\hat{x}$ we find the cosine similarity of embeddings of each pair of words and normalize by total pairs.

In the above equation, the embd is the function of finding RoBERTa embeddings of a token. In this type of summary

TABLE 2. Some summaries generated.

Reference Summary	System Summary
کرنل 001 بوکو حرام 5 کیمرون کے فوجی حالیہ لڑائی میں جاں بحق ذرائع کیمرون کی جیلوں میں سینکڑوں مشتبہ عسکریت پسند قید ہیں نائیجر نے گروپ سے لڑنے والے دوسروں کے ساتھ اتحاد کے بعد سے بوکو حرام کے مزید حملوں کا سامنا کیا ہے	فرانس کے وزیر اعظم رجب طیب اردگان گروپ نے ایک حکومتی فوجی بوکو حرام کے خلاف احتجاج کیا ہے بوکو حرام کے عسکریت پسندوں نے منگل کو ہلاکتوں کی تعداد بڑھانے کا مطالبہ کیا ہے بوکو حرام حکام کا کہنا ہے کہ ان حملوں کی ہلاکتوں کے لیے اجتماعی قبر سے 002 ہلاکتوں کی تعداد ہے
پولیس سے آزاد شوہر نے کہا کہ وہ بیوی کے حمل کے بارے میں کچھ نہیں جانتا شمالی فرانس کے گاؤں میں آٹھ شیر خوار بچوں کی لاشیں ملی ہیں جن پر رضا کارانہ قتل کا الزام ہے	نئی لاشیں ملی ہیں کیونکہ وہ اپنے شوہر کے ساتھ لڑائی کے بعد مر گئے تھے۔ پولیس نے کہا کہ وہ اسے اپنے جسم فروشی کے خلاف ورزیوں کے سلسلے میں گرفتار ہونے والی خاتون گرفتار ہونے والی خاتون کی ماں نے کہا کہ وہ جنوبی فرانس کے ساتھ زیادتی کے صوبے میں پولیس کی تحقیقات
را کی پہاڑوں کی نئی بلندیوں کو سب سے زیادہ سختی سے مارنے کی توقع ہے کچھ علاقوں میں 15 انچ تک برف پڑ سکتی ہے نیو میکسیکو میں سڑکیں برفیلی ہیں منگل کو برفانی طوفان کی وارننگ کولوراڈو سے ٹیکساس تک پھیل گئی	را کی پہاڑوں کی نئی بلندیوں کو ایک عظیم سفید سڑک پر سب سے زیادہ درجہ بندی کرتی ہے کچھ سب سے زیادہ نیو اورلینز کے ہیں جن میں سے کچھ رہائشیوں کو مارنے کی اجازت دی گئی ہے۔
8002 کے فاتح کرسٹیانو رونالڈو کو فیفا ورلڈ پلیئر آف دی ایئر کی نامزدگیوں میں شامل کیا گیا ہے ریئل میڈرڈ پانچ کھلاڑیوں میں سے ایک کو شارٹ لسٹ پر بھیجتا ہے جن میں سے کوئی بھی میڈرڈ یا بارسلونا کے ساتھ ہے لیونل میسی حال ہی میں بیلون ڈی یا ایوارڈ جیتنے کے بعد گونگ پر اترنا پسند کرتا ہے۔ یورپی پلیئر آف دی ایئر۔	کرسٹیانو رونالڈو نے اتوار کے روز سمر کیمپ میں سرفہرست رکھا تھا ، ورلڈ کپ جیتنے والے پہلے سے مایوس کن ریٹنگ والے نئے سال کی عمر میں آل سٹار ٹریک ریکارڈ کے لیے مائیکل اسٹیل کے کھلاڑیوں کو لگتا ہے کہ وہ بارسلونا کے خلاف اپیل کرے گی کیونکہ پورٹو ڈوپنگ نے بارسلونا کے ساتھ وقت فرانسیسی کل کے ساتھ وقت دائر کیا ہے۔

evaluation, we don't need to deal with the stem words or to check synonyms because the context awared embeddings already contains these information. The embedding of a word and its synonym will most likely to be very close to each other and similarity measure will be high for those words.

So the overall measure of summary evaluation by Context Awared RoBERTa Score will be the normalized similarity measure divided by the disconnection rate.

$$CARobERTaScore = \alpha \times similarity(x, \hat{x}) - \beta \times DR(\hat{x}) \quad (6)$$

where α and β are the two parameters for assigning weight to similarity index. There is trade-off between the two factors and 0.33 and 0.67 value is suitable. Average CA RoBERTa score 20.61 was measured at the examples of test-set. Studies have proved that this evaluation metric has strong correlation with the human judgements. It is also robust and is language independent. Further studies can be conducted to find the

TABLE 3. Comparison of results.

Evaluation Metric / Model	State of the art Model [128]	Transformer with One Hot encoded embeddings	Transformer with RoBERTa embeddings
Dataset	**English**	**Urdu**	**Urdu**
R-1 Score	39.92	23.27	25.18
R-2 Score	17.65	10.72	12.14
R-L Score	36.71	21.49	21.50
CA Roberta Score	27.62	17.58	20.61

optimum value of α and β, but these values can be best modeled by getting inputs from human experts.

VI. RESULTS

For better generalization of model, we need a huge amount of data but 50 training examples are not sufficient. Before this work we had only one labelled dataset of Urdu abstractive summary. We prepared dataset of more than nineteen thousand labelled examples. Experimentation was done on GPU enabled machine. Leaning curve is given in Fig. 7.

Results have been reported in $ROUGE_{Precision} score$ because we can compare the results with state of the art methods and discuss those. For Urdu language the whole pipeline of abstractive summarization has been implemented by us. Built-in libraries and functions don't support the Urdu text due to the reasons already discussed. Results have been reported by implementing our own functions for calculation of precision, recall and comparisons. The results have been reported on average ROUGE-1 score of examples of test score. State of the art ROUGE-1 score of [120] is 39.92 while our is 27.18. Figure 3 represents the difference of our and state of the art model in terms of ROUGE-1 score.

The results of summary have been reported by using transformers with positional vectors and contextual embedded vectors. Fig 8 shows the result when positional vector embeddings are used and RoBERTa contextual embeddings of Urdu language words.

The proposed evaluation metric introduces a novel approach applicable to existing methodologies. Our experimentation involved utilizing an English test dataset, where system-generated summaries underwent evaluation using the Context-Aware Roberta Score in comparison to ground truth summaries. The Roberta Embeddings were extracted from a pre-trained English language model. The introduction of context flow verification in consecutive summary sentences resulted in the Context-Aware Roberta Score consistently being lower than the ROUGE-1 score across all models and both the languages.

Table 3 shows the results of comparison of ROUGE-1, ROUGE-2 and ROUGE-L score. Although ROUGE score of Urdu is less than the English language but there is no work done for the abstractive summarization of Urdu. This is the very first work being done for the Urdu abstractive summarization. The reason of having so much gap is due to no availability of ready resources of Urdu language. Even embeddings are being optimized and for every task to code by ourselves. Even we can't find ROUGE score by any built-in function of any library. These are the baselines results, and further research can be conducted for low resources languages. ROUGE-2 score is usually less than ROUGE-1 score because in abstractive summary, two same words might not probably exist together. It calculates the recall of both words as dependent features of each other.

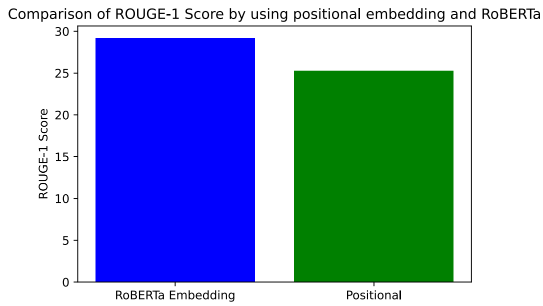

FIGURE 8. Comparison of ROUGE-1 score.

VII. CONCLUSION

There is significant difference between the results of our model and of English language model. There needs a lot of research to make the resources for low resource languages. Many journals and conferences are targeting Asian and Low Resource languages to make the life of these people better. We have tried to contribute for making resources available for Urdu language. We have prepared dataset of Urdu text summarization. Its data can not be scrapped from web resources, because a document and its summary is not given of Urdu. So we have utilized our efforts to make its dataset. We also proposed a pipeline for generating the abstractive summary of Urdu language and achieved ROUGE-1 score of 23.27. There were also limitations of the evaluation metric used for the generated summary. We proposed an evaluation metric which incorporates the context of two consecutive sentences of a summary. This evaluation metric has higher correlation between human judgement and the value of proposed evaluation metric.

VIII. FUTURE WORK

In Urdu language, a significant research is being done. Still Urdu lack ready resources. In every problem related to Urdu language, a formatted and pre-processed data is needed. We can enhance the dataset of summarization. We can contribute to both dataset of abstractive and extractive summarization. We can train a best set of embeddings and a rich vocabulary. Diacritic marks are always given a least priority in solving any problem. These marks play a vital role in understanding the language. This open a research dimension in NLP. Usually summary is 20 to 25 percent of total text document. A dataset with summary length can be made and a summarization model with optional length can be trained.

REFERENCES

[1] M. W. Bhatti and M. Aslam, "ISUTD: Intelligent system for Urdu text de-summarization," in *Proc. Int. Conf. Eng. Emerg. Technol. (ICEET)*, Feb. 2019, pp. 1–5.

[2] A. Alomari, N. Idris, A. Q. M. Sabri, and I. Alsmadi, "Deep reinforcement and transfer learning for abstractive text summarization: A review," *Comput. Speech Lang.*, vol. 71, Jan. 2022, Art. no. 101276, doi: 10.1016/j.csl.2021.101276.

[3] S.-L. Hou, X.-K. Huang, C.-Q. Fei, S.-H. Zhang, Y.-Y. Li, Q.-L. Sun, and C.-Q. Wang, "A survey of text summarization approaches based on deep learning," *J. Comput. Sci. Technol.*, vol. 36, no. 3, pp. 633–663, Jun. 2021.

[4] G. Sun, Z. Wang, and J. Zhao, "Automatic text summarization using deep reinforcement learning and beyond," *Inf. Technol. Control*, vol. 50, no. 3, pp. 458–469, Sep. 2021.

[5] W. S. El-Kassas, C. R. Salama, A. A. Rafea, and H. K. Mohamed, "Automatic text summarization: A comprehensive survey," *Expert Syst. Appl.*, vol. 165, Mar. 2021, Art. no. 113679, doi: 10.1016/j.eswa.2020.113679.

[6] A. A. Syed, F. L. Gaol, and T. Matsuo, "A survey of the state-of-the-art models in neural abstractive text summarization," *IEEE Access*, vol. 9, pp. 13248–13265, 2021.

[7] H. Jang and W. Kim, "Reinforced abstractive text summarization with semantic added reward," *IEEE Access*, vol. 9, pp. 103804–103810, 2021.

[8] A. R. Fabbri, W. Kryciski, B. McCann, C. Xiong, R. Socher, and D. Radev, "SummEval: Re-evaluating summarization evaluation," *Trans. Assoc. Comput. Linguistics*, vol. 9, pp. 391–409, Apr. 2021.

[9] H. Nanba, T. Hirao, T. Fukushima, and M. Okumura, *Text Summarization Challenge: An Evaluation Program for Text Summarization*. Cham, Switzerland: Springer, 2021.

[10] Y. Chen, C. Yang, Z. Liu, and M. Sun, "Reinforced zero-shot cross-lingual neural headline generation," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 28, no. 1 pp. 2572–2584, May 2020.

[11] W. Zhao, M. Peyrard, F. Liu, Y. Gao, C. M. Meyer, and S. Eger, "MoverScore: Text generation evaluating with contextualized embeddings and Earth mover distance," in *Proc. Conf. Empirical Methods Natural Language Process. 9th Int. Joint Conf. Natural Language Process. (EMNLP-IJCNLP)*, 2019, pp. 563–578.

[12] Y. Zou, X. Zhang, W. Lu, F. Wei, and M. Zhou, "Pre-training for abstractive document summarization by reinstating source text," 2020, *arXiv:2004.01853*.

[13] N. Stiennon, L. Ouyang, J. Wu, D. M. Ziegler, R. Lowe, C. Voss, A. Radford, D. Amodei, and P. Christiano, "Learning to summarize from human feedback," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2022, pp. 1–45.

[14] F. Sheng Bao, H. Li, G. Luo, M. Qiu, Y. Yang, Y. He, and C. Chen, "SueNes: A weakly supervised approach to evaluating single-document summarization via negative sampling," 2020, *arXiv:2005.06377*.

[15] W. Kryciski, R. Paulus, C. Xiong, and R. Socher, "Improving abstraction in text summarization," in *Proc. Conf. Empirical Methods Natural Language Process.*, 2018, pp. 1808–1817.

[16] A. Nawaz, M. Bakhtyar, J. Baber, I. Ullah, W. Noor, and A. Basit, "Extractive text summarization models for Urdu language," *Inf. Process. Manage.*, vol. 57, no. 6, Nov. 2020, Art. no. 102383, doi: 10.1016/j.ipm.2020.102383.

[17] A. Khattak, M. Z. Asghar, A. Saeed, I. A. Hameed, S. A. Hassan, and S. Ahmad, "A survey on sentiment analysis in Urdu: A resource-poor language," *Egyptian Informat. J.*, vol. 22, no. 1, pp. 53–74, Mar. 2021, doi: 10.1016/j.eij.2020.04.003.

[18] P. Janjanam and C. P. Reddy, "Text summarization: An essential study," in *Proc. Int. Conf. Comput. Intell. Data Sci. (ICCIDS)*, Feb. 2019, pp. 1–6.

[19] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "BERTScore: Evaluating text generation with BERT," 2019, *arXiv:1904.09675*.

[20] M. A. Mosa, A. S. Anwar, and A. Hamouda, "A survey of multiple types of text summarization with their satellite contents based on swarm intelligence optimization algorithms," *Knowl.-Based Syst.*, vol. 163, pp. 518–532, Jan. 2019, doi: 10.1016/j.knosys.2018.09.008.

[21] J. Zhang, K. Jia, J. Jia, and Y. Qian, "An improved approach to infer protein–protein interaction based on a hierarchical vector space model," *BMC Bioinf.*, vol. 19, no. 1, pp. 1–9, Dec. 2018.

[22] R. Xu, C. Li, A. H. Paterson, Y. Jiang, S. Sun, and J. S. Robertson, "Aerial images and convolutional neural network for cotton Bloom detection," *Frontiers Plant Sci.*, vol. 8, pp. 1–17, Feb. 2018.

[23] D. Minnen, G. Toderici, M. Covell, T. Chinen, N. Johnston, J. Shor, S. J. Hwang, D. Vincent, and S. Singh, "Spatially adaptive image compression using a tiled deep network," 2018, *arXiv:1802.02629*.

[24] E. Agustsson, M. Tschannen, F. Mentzer, R. Timofte, and L. Van Gool, "Generative adversarial networks for extreme learned image compression," 2018, *arXiv:1804.02958*.

[25] L. Wan, "Extraction algorithm of English text summarization for English teaching," in *Proc. 3rd Int. Conf. Intell. Transp.*, 2018, pp. 307–310.

[26] J. Krantz and J. Kalita, "Abstractive summarization using attentive neural techniques," 2018, *arXiv:1810.08838*.

[27] R. Paulus, C. Xiong, and R. Socher, "A deep reinforced model for abstractive summarization," in *Proc. 6th Int. Conf. Learn. Represent.*, 2018, pp. 1–12.

[28] A. Muhammad, N. Jazeb, A. M. Martinez-Enriquez, and A. Sikander, "EUTS: Extractive Urdu text summarizer," in *Proc. 17th Mex. Int. Conf. Artif. Intell. (MICAI)*, Oct. 2018, pp. 39–44.

[29] A. G. Ororbia, A. Mali, J. Wu, S. O'Connell, D. Miller, and C. Lee Giles, "Learned neural iterative decoding for lossy image compression systems," 2018, *arXiv:1803.05863*.

[30] L. Zhao, H. Bai, F. Li, A. Wang, and Y. Zhao, "Mixed-resolution image representation and compression with convolutional neural networks," 2018, *arXiv:1802.01447*.

[31] X. Sun and H. Zhuge, "Summarization of scientific paper through reinforcement ranking on semantic link network," *IEEE Access*, vol. 6, pp. 40611–40625, 2018.

[32] K. Merchant and Y. Pande, "NLP based latent semantic analysis for legal text summarization," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Sep. 2018, pp. 1803–1807.

[33] K.-Y. Chen, S.-H. Liu, B. Chen, and H.-M. Wang, "An information distillation framework for extractive summarization," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 26, no. 1, pp. 161–170, Jan. 2018.

[34] S. N. Sayyed, A. M. Pundge, and C. N. Mahender, "Text mining for proper identification of noun using conceptual dependency," in *Proc. Int. Conf. Smart Syst. Inventive Technol. (ICSSIT)*, Dec. 2018, pp. 463–467.

[35] K. Shakeel, G. R. Tahir, I. Tehseen, and M. Ali, "A framework of Urdu topic modeling using Latent Dirichlet allocation (LDA)," in *Proc. IEEE 8th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Jan. 2018, pp. 117–123.

[36] J. Tan, X. Wan, and J. Xiao, "Abstractive document summarization with a graph-based attentional neural model," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, 2017, pp. 1171–1181.

[37] S. S. Grewal, "Full resolution image compression with deep neural networks," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 5, no. 6, pp. 1125–1134, Sep. 2017.

[38] (2017). *RxNLP What is ROUGE and How It Works for Evaluation of Summarization Tasks?*. [Online]. Available: http://www.rxnlp.com/how-rouge-works-for-evaluation-of-summarization-tasks/

[39] G. Lee and K. Lee, "Automatic text summarization using reinforcement learning with embedding features," in *Proc. 8th Int. Joint Conf. Natural Language Process.*, 2017, pp. 193–197.

[40] M. Humayoun, R. Nawab, M. Uzair, S. Aslam, and O. Farzand, "Urdu summary corpus," in *Proc. 10th Int. Conf. Language Resour. and Eval.*, 2016, pp. 796–800.
[41] R. Nallapati, B. Zhou, C. dos Santos, C. Gulcehre, and B. Xiang, "Abstractive text summarization using sequence-to-sequence RNNs and beyond," in *Proc. 20th SIGNLL Conf. Comput. Natural Lang. Learn.*, 2016, pp. 280–290.
[42] E. Turajli, E. Engineering, and S. Sarajevo, "Application of neural networks to compression of CT images," in *Proc. Int. Symp. Telecommun.*, 2016, pp. 1–6.
[43] M. Denkowski and A. Lavie, "Meteor universal: Language specific translation evaluation for any target language," in *Proc. 9th Workshop Stat. Mach. Transl.*, 2014, pp. 376–380.
[44] J. Wei, "Application of hybrid back propagation neural network in image compression," in *Proc. 8th Int. Conf. Intell. Comput. Technol. Autom. (ICICTA)*, Jun. 2015, pp. 209–212.
[45] K.-Y. Chen, S.-H. Liu, B. Chen, H.-M. Wang, E.-E. Jan, W.-L. Hsu, and H.-H. Chen, "Extractive broadcast news summarization leveraging recurrent neural network language modeling techniques," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 23, no. 8, pp. 1322–1334, Aug. 2015.
[46] S. C. Roca, "Automatic text summarization," *Digithum*, vol. 37, no. 3, pp. 1–348, Sep. 2001.
[47] M. T. Mills and N. G. Bourbakis, "Graph-based methods for natural language processing and understanding—A survey and analysis," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 44, no. 1, pp. 59–71, Jan. 2014.
[48] S. Mishra and S. Savarkar, "Image compression using neural network," in *Proc. Int. Conf. Workshop Emerg. Trends Technol.*, 2012, pp. 18–21.
[49] M. G. Ozsoy, F. N. Alpaslan, and I. Cicekli, "Text summarization using latent semantic analysis," *J. Inf. Sci.*, vol. 37, no. 4, pp. 405–417, Aug. 2011.
[50] J. S. Kallimani, K. G. Srinivasa, and E. Reddy B, "Information extraction by an abstractive text summarization for an Indian regional language," in *Proc. 7th Int. Conf. Natural Language Process. Knowl. Eng.*, Nov. 2011, pp. 319–322.
[51] M. S. Binwahlan, N. Salim, and L. Suanmali, "Fuzzy swarm diversity hybrid model for text summarization," *Inf. Process. Manage.*, vol. 46, no. 5, pp. 571–588, Sep. 2010, doi: 10.1016/j.ipm.2010.03.004.
[52] M. S. Binwahlan, N. Salim, and L. Suanmali, "Fuzzy swarm based text summarization," *J. Comput. Sci.*, vol. 5, no. 5, pp. 338–346, May 2009.
[53] M. S. Binwahlan, N. Salim, and L. Suanmali, "Swarm based text summarization," in *Proc. Int. Assoc. Comput. Sci. Inf. Technol. Spring Conf.*, Apr. 2009, pp. 145–150.
[54] M. A. Fattah, "Automatic text summarization," *Digithum*, vol. 37, no. 3, pp. 1–348, Sep. 2001.
[55] A. Niewiadomski, "A Type-2 fuzzy approach to linguistic summarization of data," *IEEE Trans. Fuzzy Syst.*, vol. 16, no. 1, pp. 198–212, Feb. 2008.
[56] T. Fatima, R. Islam, and M. Anwar, "Morphological and orthographic challenges in Urdu language processing: A Review," in *Proc. 13th Workshop Asian Lang. Resour.*, no. 1, 2007, pp. 44–51. [Online]. Available: http://lrec-conf.org/workshops/lrec2018/W29/
[57] S. Yong, A. Abidin, and Y. Chen, "A neural-based text summarization system," *WIT Trans. Inf. Commun. Technol.*, vol. 37, pp. 185–192, 2006. [Online]. Available: https://www.witpress.com/elibrary/wit-transactions-on-information-and-communication-technologies/37/16716
[58] S.-T. Yuan and J. Sun, "Ontology-based structured cosine similarity in document summarization: With applications to mobile audio-based knowledge management," *IEEE Trans. Syst. Man Cybern., Part B (Cybernetics)*, vol. 35, no. 5, pp. 1028–1040, Oct. 2005.
[59] H. Jing, "Sentence reduction for automatic text summarization," in *Proc. 6th Conf. Appl. Natural Language Process.*, 2000, pp. 1–7.
[60] G. Wentzel, "Funkenlinien Im R´ontgenspektrum," *Annalen Der Phys.*, vol. 371, no. 1, pp. 437–461, 1992.
[61] T. Wolf et al., "Transformers: State-of-the-art natural language processing," in *Proc. Conf. Empirical Methods Natural Language Process, Syst. Demonstrations*, 2020, pp. 38–45.
[62] P. Laban, A. Hsi, J. Canny, and M. A. Hearst, "The summary loop: Learning to write abstractive summaries without examples," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 5135–5150.
[63] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
[64] C. Lin, "Rouge: A package for automatic evaluation of summaries," *Text Summarization Branches Out*, pp. 74–81, 2004. [Online]. Available: https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=Rouge%3A+A+package+for+automatic+evaluation+of+summaries&btnG=
[65] S. Maples, "The ROUGE-AR: A proposed extension to the Rouge evaluation metric for abstractive text summarization," *Symbolic Syst. Dept.*, 2017.
[66] K. Ganesan, "ROUGE 2.0: Updated and improved measures for evaluation of summarization tasks," 2018, *arXiv:1803.01937*.
[67] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: A method for automatic evaluation of machine translation," in *Proc. 40th Annu. Meeting Assoc. Comput. Linguistics*, 2001, pp. 311–318.
[68] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open source toolkit for statistical machine translation," in *Proc. 45th Annu. Meeting Assoc. Comput. Linguistics Companion*, 2007, pp. 177–180.
[69] M. Denkowski and A. Lavie, "Meteor universal: Language specific translation evaluation for any target language," in *Proc. 9th Workshop Stat. Mach. Transl.*, 2014, pp. 376–380.
[70] Y. Guo and J. Hu, "Meteor++ 2.0: Adopt syntactic level paraphrase knowledge into machine translation evaluation," in *Proc. 4th Conf. Mach. Transl.*, 2019, pp. 501–506.
[71] G. Doddington, "Automatic evaluation of machine translation quality using N-gram Co-occurrence statistics," in *Proc. 2nd Int. Conf. Hum. Language Technol. Res.*, 2002, pp. 138–145.
[72] M. Galley, C. Brockett, A. Sordoni, Y. Ji, M. Auli, C. Quirk, M. Mitchell, J. Gao, and B. Dolan, "DeltaBLEU: A discriminative metric for generation tasks with intrinsically diverse targets," 2015, *arXiv:1506.06863*.
[73] M. Popović, "ChrF: Character N-gram F-score for automatic MT evaluation," in *Proc. 10th Workshop Stat. Mach. Transl.*, 2015, pp. 392–395.
[74] M. Popović, "ChrF++: Words helping character N-grams," in *Proc. 2nd Conf. Mach. Transl.*, 2017, pp. 612–618.
[75] R. Vedantam, C. L. Zitnick, and D. Parikh, "CIDEr: Consensus-based image description evaluation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 4566–4575.
[76] A. Tejasvi Chaganty, S. Mussman, and P. Liang, "The price of debiasing automatic metrics in natural language evaluation," 2018, *arXiv:1807.02202*.
[77] T. B. Hashimoto, H. Zhang, and P. Liang, "Unifying human and statistical evaluation for natural language generation," 2019, *arXiv:1904.02792*.
[78] V. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Sov. Phys. Doklady*, vol. 10, no. 8, pp. 707–710, 1996.
[79] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul, "A study of translation edit rate with targeted human annotation," in *Proc. 7th Conf. Assoc. Mach. Transl. Americas, Tech. Papers*, 2006, pp. 223–231.
[80] J. Panja and S. K. Naskar, "ITER: Improving translation edit rate through optimizable edit costs," in *Proc. 3rd Conf. Mach. Transl.:,Shared Task Papers*, 2018, pp. 746–750.
[81] C. Tillmann, S. Vogel, H. Ney, A. Zubiaga, and H. Sawaf, "Accelerated DP based search for statistical translation.," *Eurospeech*, 1997.
[82] G. Leusch, N. Ueffing, and H. Ney, "Efficient MT evaluation using block movements," in *Proc. EACL-1th Conf. Eur. Chapter Assoc. Comput. Linguistics*, 2006, pp. 241–248.
[83] W. Wang, J.-T. Peter, H. Rosendahl, and H. Ney, "CharacTer: Translation edit rate on character level," in *Proc. 1st Conf. Mach. Transl.*, 2016, pp. 505–510.
[84] P. Stanchev, W. Wang, and H. Ney, "EED: Extended edit distance measure for machine translation," in *Proc. 4th Conf. Mach. Transl.*, 2019, pp. 514–520.
[85] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.
[86] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*, 2014, pp. 1532–1543.
[87] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov, "Learning word vectors for 157 languages," 2018, *arXiv:1802.06893*.

[88] D. Quoc Nguyen, D. Quoc Nguyen, A. Modi, S. Thater, and M. Pinkal, "A mixture model for learning multi-sense word embeddings," 2017, *arXiv:1706.05111*.
[89] B. Athiwaratkun, A. Gordon Wilson, and A. Anandkumar, "Probabilistic fasttext for multi-sense word embeddings," 2018, *arXiv:1806.02901*.
[90] C.-K. Lo, "MEANT 2.0: Accurate semantic MT evaluation for any output language," in *Proc. 2nd Conf. Mach. Transl.*, 2017, pp. 589–597.
[91] C.-K. Lo, M. Simard, D. Stewart, S. Larkin, C. Goutte, and P. Littell, "Accurate semantic textual similarity for cleaning noisy parallel corpora using semantic machine translation evaluation metric: The NRC supervised submissions to the parallel corpus filtering task," in *Proc. 3rd Conf. Mach. Transl., Shared Task Papers*, 2018, pp. 908–916.
[92] C. Corley and R. Mihalcea, "Measuring the semantic similarity of texts," in *Proc. ACL Workshop Empirical Model. Semantic Equivalence Entailment*, 2005, pp. 13–18.
[93] M. Kusner, Y. Sun, N. Kolkin, and K. Weinberger, "From word embeddings to document distances," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 957–966.
[94] J. Chow, L. Specia, and P. Madhyastha, "WMDO: Fluency-based word Mover's distance for machine translation evaluation," in *Proc. 4th Conf. Mach. Transl.*, 2019, pp. 494–500.
[95] E. Clark, A. Celikyilmaz, and N. A. Smith, "Sentence Mover's similarity: Automatic evaluation for multi-sentence texts," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 2748–2760.
[96] Y. Rubner, C. Tomasi, and L. J. Guibas, "A metric for distributions with applications to image databases," in *Proc. 6th Int. Conf. Comput. Vis.*, 1998, pp. 59–66.
[97] V. Rus and M. Lintean, "An optimal assessment of natural language student input using word-to-word similarity metrics," in *Proc. Int. Conf. Intell. Tutoring Syst.*, 2012, pp. 675–676.
[98] S. Sharma, L. El Asri, H. Schulz, and J. Zumer, "Relevance of unsupervised metrics in task-oriented dialogue for evaluating natural language generation," 2017, *arXiv:1706.09799*.
[99] M. Stanojevic and K. Sima'an, "BEER: BEtter evaluation as ranking," in *Proc. 9th Workshop Stat. Mach. Transl.*, 2014, pp. 414–419.
[100] Q. Ma, Y. Graham, S. Wang, and Q. Liu, "Blend: A novel combined MT metric based on direct assessment—CASICT-DCU submission to WMT17 metrics task," in *Proc. 2nd Conf. Mach. Transl.*, 2017, pp. 598–603.
[101] H. Shimanaka, T. Kajiwara, and M. Komachi, "RUSE: Regressor using sentence embeddings for automatic machine translation evaluation," in *Proc. 3rd Conf. Mach. Transl., Shared Task Papers*, 2018, pp. 751–758.
[102] Y. Cui, G. Yang, A. Veit, X. Huang, and S. Belongie, "Learning to evaluate image captioning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5804–5812.
[103] R. Lowe, M. Noseworthy, I. V. Serban, N. Angelard-Gontier, Y. Bengio, and J. Pineau, "Towards an automatic Turing test: Learning to evaluate dialogue responses," 2017, *arXiv:1708.07149*.
[104] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.
[105] J. Sarzynska-Wawer, A. Wawer, A. Pawlak, J. Szymanowska, I. Stefaniak, M. Jarkiewicz, and L. Okruszek, "Detecting formal thought disorder by deep contextualized word representations," *Psychiatry Res.*, vol. 304, Oct. 2021, Art. no. 114135.
[106] A. Lavie and A. Agarwal, "Meteor: An automatic metric for MT evaluation with high levels of correlation with human judgments," in *Proc. 2nd Workshop Stat. Mach. Transl.*, 2007, pp. 65–72.
[107] Y. Goldberg and O. Levy, "Word2vec explained: Deriving Mikolov et al.'s negative-sampling word-embedding method," 2014, *arXiv:1402.3722*.
[108] B. Hu, Q. Chen, and F. Zhu, "LCSTS: A large scale Chinese short text summarization dataset," 2015, *arXiv:1506.05865*.
[109] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," 2017, *arXiv:1704.04368*.
[110] S. Gehrmann, Y. Deng, and A. M. Rush, "Bottom-up abstractive summarization," 2018, *arXiv:1808.10792*.
[111] C. Zhu, R. Xu, M. Zeng, and X. Huang, "End-to-end abstractive summarization for meetings," 2020, *arXiv:2004.02016*.
[112] Y. Cao, X. Wan, J. Yao, and D. Yu, "MultiSumm: Towards a unified model for multi-lingual abstractive summarization," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 11–18.
[113] W. Kryciski, R. Paulus, C. Xiong, and R. Socher, "Improving abstraction in text summarization," 2018, *arXiv:1808.07913*.
[114] W. Li, X. Xiao, Y. Lyu, and Y. Wang, "Improving neural abstractive document summarization with explicit information selection modeling," in *Proc. Conf. Empirical Methods Natural Language Process.*, 2018, pp. 1787–1796.
[115] L. Perez-Beltrachini, Y. Liu, and M. Lapata, "Generating summaries with topic templates and structured convolutional decoders," 2019, *arXiv:1906.04687*.
[116] A. Cohan, F. Dernoncourt, D. Soon Kim, T. Bui, S. Kim, W. Chang, and N. Goharian, "A discourse-aware attention model for abstractive summarization of long documents," 2018, *arXiv:1804.05685*.
[117] Y.-C. Chen and M. Bansal, "Fast abstractive summarization with reinforce-selected sentence rewriting," 2018, *arXiv:1805.11080*.
[118] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*.
[119] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*.
[120] L. Liu, Y. Lu, M. Yang, Q. Qu, J. Zhu, and H. Li, "Generative adversarial network for abstractive text summarization," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 1–2.
[121] G. Vilca and M. Cabezudo, "A study of abstractive summarization using semantic representations and discourse level information," in *Lecture Notes In Computer Science*, 2017, pp. 482–490.
[122] A. Jabbar, S. Iqbal, M. U. G. Khan, and S. Hussain, "A survey on Urdu and Urdu like language stemmers and stemming techniques," in *Proc. 20th Int. Conf. Text, Speech, Dialogue (TSD)*. Prague, Czech Republic: Springer, Aug. 2017, pp. 482–490.
[123] N. Durrani and S. Hussain, "Urdu word segmentation," in *Proc. Hum. Language Techn. Annu. Conf. North Amer. Assoc. Comput. Linguistics, Main Conf.*, 2010, pp. 528–536.
[124] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, 2017, pp. 1073–1083.
[125] P. Verma, A. Verma, and S. Pal, "An approach for extractive text summarization using fuzzy evolutionary and clustering algorithms," *Appl. Soft Comput.*, vol. 120, May 2022, Art. no. 108670.
[126] B. Mutlu, E. A. Sezer, and M. A. Akcayol, "Candidate sentence selection for extractive text summarization," *Inf. Process. Manage.*, vol. 57, no. 6, Nov. 2020, Art. no. 102359.

HASSAN RAZA received the B.S. degree in computer science from COMSATS University Islamabad, in 2016, and the M.S. degree in computer science from the National University of Computer and Emerging Sciences, Islamabad, Pakistan, in 2019. He is currently pursuing the Ph.D. degree in computer science with the National University of Computer and Emerging Sciences. He was a Researcher with the Recognition, Vision, and Learning Research Laboratory (ReVeaL). He has been working in the domain of data mining, machine learning, deep learning, computer vision, and natural language processing, for the last six years. He has also worked on different projects, including the Demand Supply Gap Prediction for CAB Hiring Systems to Minimize Waiting Time and For Better User Experience. He is working with the Artificial Intelligence and Machine Learning Laboratory (AIM Lab). His research work involves information extraction and processing in low-resource environments and text mining and summarization techniques. His teaching interests include artificial intelligence and object-oriented programming.

WASEEM SHAHZAD received the B.S. degree in math stat from Panjab University, in 2000, the M.Sc. degree in IT, in 2004, and the M.S. and Ph.D. degrees in computer science from the National University of Computer and Emerging Sciences, Islamabad, Pakistan, in 2008. He is currently the Campus Director of the National University of Computer and Emerging Sciences. He is also a Founding Member with the Artificial Intelligence and Machine Learning Laboratory (AIM). His research work involves data mining, computational intelligence, machine learning, and theory of computation. He has various research publications in these fields.