International Journal on Semantic Web and Information Systems
Volume 18 • Issue 1

Modified Transformer Architecture to Explain Black Box Models in Narrative Form

Diksha Malhotra, Punjab Engineering College, India*
Poonam Saini, Punjab Engineering College, India
Awadhesh Kumar Singh, National Institute of Technology, Kurukshetra, India

ABSTRACT

The current XAI techniques present explanations mainly as visuals and structured data. However, these explanations are difficult to interpret for a non-expert user. Here, the use of natural language generation (NLG)-based techniques can help to represent explanations in a human-understandable format. The paper addresses the issue of automatic generation of narratives using a modified transformer approach. Further, due to the unavailability of a relevant annotated dataset for development and testing, the authors also propose a verbalization template approach to generate the same. The input of the transformer is linearized to convert the data-to-text task into text-to-text task. The proposed work is evaluated on a verbalized explained PIMA Indians diabetes dataset and exhibits significant improvement as compared to existing baselines for both manual and automatic evaluation. Also, the narratives provide better comprehensibility to be trusted by human evaluators than the non-NLG counterparts. Lastly, an ablation study is performed in order to understand the contribution of each component.

KEYWORDS

Data-to-Text, Explainable Artificial Intelligence, LIME, Natural Language Generation, Neural Network, SVM, Template Generation, Transformer

INTRODUCTION

With the advancement in technology, Artificial Intelligence (AI) has gained enormous popularity and applicability in various domains such as healthcare, finance, retail, etc. (Sarivougioukas & Vagelatos, 2020). In order to produce high-performance commercial products, many AI-based companies tend to develop predictive models whose behaviour may sometimes deviate from human expectations (Cheng et al., 2021). Traditional AI systems often lack transparency in decisions due to their complex nature and hence, are unable to explain such deviations (EU, 2019). In critical systems like autonomous (Fiorini, 2020; Pandey & Banerjee, 2019) and AI-assisted healthcare systems (Gupta et al., 2021; Sun et al., 2019), there is a need to induce explainability of decisions for social, practical, and legal reasons. Hence, recently the branch of eXplainable Artificial Intelligence (XAI) has gained importance in applications where the result of committing a mistake can be disastrous (Gunning &

DOI: 10.4018/IJSWIS.297040 *Corresponding Author

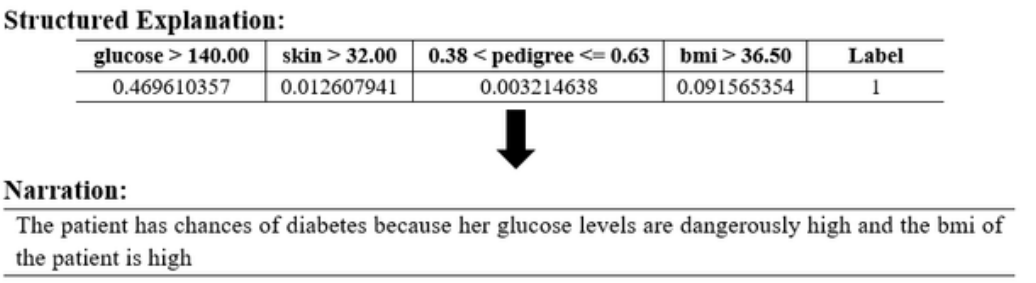
Aha, 2019). It refers to the branch of AI which provides reasoning behind the predictions of any AI model. Further, XAI techniques can be broadly divided into intrinsic and post-hoc wherein, intrinsic techniques aim to provide explanation along with the prediction. Whereas the post-hoc techniques are applied on models to produce explanation after the output is predicted. The paper aims to build an explanation-to-narration module for post-hoc explanations.

The current state-of-art XAI techniques present explanations in many forms such as *visual, audio, linguistic, tabular*, etc. The traditional trend in the literature is to represent results in the form of visuals, especially heat maps. However, these may not be well understood by a non-technical user. Out of the above-mentioned ways, linguistic methods can be attractive for interested non-expert users (J.M. Alonso et al., 2020). These allow users to understand the model's predictions without any mathematics or engineering background and instigate willingness among them to use autonomous systems (J.M. Alonso et al., 2020). To date, few works have directly addressed the possibility of generating textual explanations from the structured output of an explainer. However, the NLP community (Singh & Sachan, 2021), especially working in data-to-text generation, can add a linguistic layer to many of the state-of-art post-hoc XAI systems proposed so far (Fayoumi & Hajjar, 2020; Inan & Dikenelli, 2021).

The explanations generated by the state-of-art post-hoc local XAI techniques such as *LIME* (Ribeiro et al., 2016), *SHAP* (Roth, 1988), etc. are generally in the form of $S_i = \left(\{FC_j, F_j\}; \forall j \in \left(0, n\right)\right)$ where F_j and FC_j represent the features and their contribution respectively for each instance i. Natural Language Generation (NLG) techniques can convert data into text or text into text depending on the application requirement (Reiter & Dale, 1997). A sub-field of NLG *i.e., data-to-text* generation can be employed on the structured explanation $\left(S_i\right)$ to generate the corresponding narrative.

Traditionally, NLG uses templates to generate text from the given structured data. Although the template-based approach offers high linguistic quality and seamless content, it requires manual effort and is not diverse in nature. This static nature of templates lacks stylistic variation and somehow produces non-natural sentences. However, neural networks can help in generalizing beyond a limited amount of annotated data or templates. In this paper, the authors present a neural model for explanation-to-narrative generation by extending a transformer-based model (Vaswani et al., 2017) that was formerly developed for the text-to-text generation task. One of the main challenges while proposing such a model for explanation-to-narrative generation is the absence of an annotated dataset containing weight contributions for each feature, and their corresponding annotated narratives. To address this challenge, authors first propose a verbalizational template-based data annotation technique. The proposed model is then trained for lexicalization and linguistic realization from such a collection of explanation-narrative pairs using a modified transformer model. The resulting generated narrations are compared to some NLG baseline model outputs using automatic and human evaluations. Also,

Figure 1. An example of the structured explanation and corresponding narrations



the authors present a comparison between the generated NLG interpretation of the explanation and a structured explanation.

The main contributions for this paper are:

1. Introduce selection and template-based data annotation techniques for explanation-to-narrative generation to prepare training datasets consisting of templated narrations along with explanations.
2. Propose a modified transformer-based model to generate narratives trained on the created pairs from the dataset. To the best of knowledge, this is the first work to explore the explanation-to-narrative generation problem using the template-driven neural model.
3. Carry out a series of evaluations to compare the model's performance with existing baselines.
4. Evaluate the generated narratives on various automatic and human evaluations.

The rest of the paper is organized as follows.

Section 2 discusses the existing literature related to explanation-to-narrative and data-to-text generation and Section 3 explains, in detail, the process steps along with the modified transformer architecture for narrative generation. The dataset, model baselines and various evaluation metrics used in the work have been described in Section 4. Section 5 presents the results and analysis along with an ablation study to validate the proposed architecture followed by conclusion and future work.

RELATED WORK

The section discusses various techniques applied for explanation-to-narrative generation to date. Also, various state-of-art techniques for a data-to-text generation used in literature have been reviewed.

Explanation-to-Narrative Generation

The task of explanation-to-narrative generation can be defined as generating text corresponding to the explanation generated by an explainer for a particular machine learning prediction. As an initial attempt to produce text for the explanation generated by post-hoc (Ribeiro et al., 2016) local explainer (LIME), Forest et al. (Forrest et al., 2018) proposed a template-based approach where the explanations are converted to paragraphs using slot-value replacement approach. The authors used SimpleNLG (Gatt & Reiter, 2009) to create templates. However, the templates are static in nature, lack variation in narration, and require a lot of human intervention. Further, Alonso and Bugarin (Jose M Alonso & Bugar{\'\i}n, 2019) developed a web service, namely, *ExpliClass*, which produces post-hoc explanations for complex models in natural language using state-of-art NLG pipeline proposed by Reiter et al. (Reiter & Dale, 1997).

Besides using a template-based approach, researchers have also exploited End-to-End text generation approaches for generating textual explanations. In 2015, Xu et al. (K. Xu et al., 2015) incorporated encoder-decoder based narration module in their image classification model to semantically explain what was detected by the model. Similarly, in 2019, an explainable cancer diagnosis system was proposed to automatically produce textual reports leveraging the image caption model trained on image-pathologist report pairs (Z. Zhang et al., 2019). Authors (Park et al., 2018) also used multimodal justification pairs to generate the model prediction and corresponding textual justification. Further, authors (Ehsan et al., 2018) coined the term *AI rationalization* as a process to generate human-like explanations of the recommender system behaviour. They have proposed an LSTM encoder-decoder-based narrative generation system that accepts the states, actions, and corresponding annotations from the model. However, they don't utilize the explanations from a post-hoc explainer to generate the narratives. In this paper, authors have exploited the use of transformers to explain the post-hoc explanations in the form of narratives.

Table 1. Explanation-to-narrative generation literature

S.No.	Authors	Technique	Post-hoc/ Intrinsic Explanations	Limitations
1.	Forest et al.	Template-based approach using SimpleNLG	Post-hoc (LIME)	Template based approaches lack stylistic variability
2.	Alonso and Bugarin	NLG pipeline	Post-hoc (LIME)	The approach is slow and inefficient
3.	Xu et al.	Encoder-decoder model	Intrinsic	The approach is not generalizable
4.	Z. Zhang et al.	Encoder-decoder model	Intrinsic	The approach is not generalizable
5.	Park et al.	Multimodal justification	Intrinsic	The approach cannot be used for generating post-hoc narratives
6.	Ehsan et al.	LSTM encoder-decoder	Intrinsic	The approach is time extensive

Data-to-Text Generation

The main aim of the data-to-text generation task is to generate text corresponding to the given structured data. This section reviews various state-of-art techniques for data-to-text generation especially tabular data. Several domain-specific tasks such as weather-forecast (Reiter et al., 2005), sports game summarization, and biography generation (Lebret et al., 2016) have exploited the use of data-to-text generation. The research in data-to-text generation increased due to the availability of large datasets such as E2E (Novikova et al., 2017), ROTOWIRE (Wiseman et al., 2017) and WebNLG (Gardent et al., 2017). Authors exploited many techniques ranging from template-based (Goldberg et al., 1994; van der Lee et al., 2017) to a pipeline and further to deep learning based end-to-end approaches. Traditional techniques such as template-based and rule-based divide the task of data-to-text into two major subtasks *i.e.,* what to say (*content selection*) and how to say (*surface realization*) (Reiter & Dale, 1997). Further, several authors (Belz & Kow, 2009; Langner et al., 2010; Pereira et al., 2015) employed Statistical Machine Translation (SMT) for data-to-text generation. These techniques automatically learn and combine the two subtasks.

However, deep learning-based Neural Machine Translation (NMT) outperformed the SMT approaches (Wiseman et al., 2017). Hence, data-to-text experts have proposed to use neural models by linearizing the structured input to text. Here, the data-to-text task is transformed into a text-to-text generation task. Initially, authors (Sutskever et al., 2014a) proposed Sequence-to-Sequence (Seq2Seq) based models to convert data to text. Such systems employ encoder-decoder networks, which use recurrent neural networks (RNN) and its variants as basic units (Nie et al., 2019). Following this, authors (Bahdanau et al., 2014) proposed Seq2Seq models with attention mechanism in the encoder and decoder layers. Sometimes, these models are unable to generate some important rare words from the input vocabulary. Hence, to address this problem, (Gu et al., 2016) incorporated a copying mechanism in the Seq2Seq models to include such rare words directly from the input text to the output text. However, these models suffered from long-term dependency problems due to the recurrent units. Further, Generative Adversarial Networks (GANs) are also employed in literature to generate text. However, they do not contain an encoder unit and are difficult to train. Also, in some cases, the model may collapse and is unable to capture the real distribution of data (Tolstikhin et al., 2017). In order to tackle these issues, authors (Vaswani et al., 2017) proposed a self-attention based

Table 2. Data-to-text generation literature

S.No.	Authors	Technique	Limitations
1.	Reiter & Dale et al.	Pipelined approach	Time consuming
2.	Goldberg et al., van der Lee et al.	Template	Lacks stylistic variability
3.	Belz & Know, Langner et al., Pereira et al.	Statistical Machine Translation (SMT)	Time consuming
4.	Wiseman et al.	Neural Machine Translation (NMT)	-
5.	Bahdanau et al.	Seq2seq models with additional attention mechanism	Time consuming and does not include rare words
6.	Gu et al.	Seq2seq models with copying mechanism	Time consuming
7.	Tolstikhin et al.	Generative Adversarial Networks (GANs)	Difficult to train and may collapse
8.	X. Xu et al.	Transformers	Does not include rare words

transformer model. Initially, the transformer model was developed for text-to-text generation tasks such as summarization, machine translation, question answering, etc (Li et al., 2021; Moosavi et al., 2021; Yermakov et al., 2021) It has been found that in various data-to-text tasks (X. Xu et al., 2020), transformers perform better than Seq2Seq models.

PROPOSED NARRATIVE GENERATION (MODIFIED TRANSFORMER) MODEL

The section describes the proposed work. It first explains the problem statement, followed by the architecture of the proposed modified transformer for XAI narrative generation.

Problem Statement

This study aims to model the explanation-to-narrative generation in an encoder-decoder framework where a record of structured explanation is given as input to generate a narrative summary of the explanation.

Given an explanation in the form of a record (R) represented by feature-contribution pairs $\left\{(F_1,C_1),(F_2,C_2),\ldots,(F_i,C_i)\right\}$ for an instance, the target is to generate a narrative for the explanation. For each feature F_i contributing towards the prediction of an instance, there exists a feature contribution C_i. The narrative is, hence, a well-formed textual summary $\left(y=\left\{w_1,w_2,\ldots w_t\right\};t \ is \ the \ sentence \ length\right)$ of the record R. The paper proposes to model the explanation-to-narrative generation task as a text-to-text generation task which can be represented as:

$$\alpha = \arg\max_{\theta} \sum_{(R,y)} \log P(y \mid R;\theta) \tag{1}$$

where α is the next generated word and θ is the hyperparameter.

Proposed Architecture

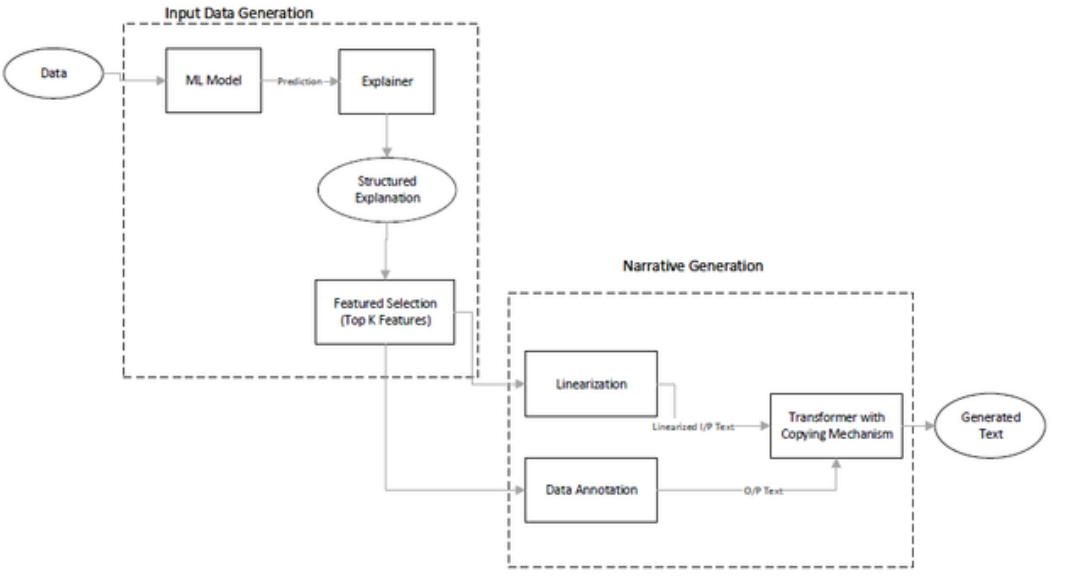
The paper employs the modified transformer-based encoder-decoder model to translate the explanations to narratives. Table 3 shows the difference between the vanilla transformer and the modified transformer. Further, figure 2 shows the detailed pipeline of the proposed model. The pipeline of the model can be broadly divided into two tasks, namely, *Input data generation* and *narrative generation*.

The post-hoc local explanation for the prediction of a particular instance (i) is in the form of a record $\left(FC_i, FQ_i\right)$. The explainer discretizes each feature into quantiles $\left(FQ_i\right)$ and assigns the contribution value $\left(FC_i\right)$ to each FQ_i associated with the instance. As positive contributions are significant for positive labels and negative contributions for negative labels, hence, the data cleaning and mapping step prunes the insignificant values and maps the features with the contributions. Further, a verbalizational template is generated to facilitate the data annotation step whose output can be used as reference text for the proposed transformer model. Also, the structured explanations are linearized to be fed as input to the transformer. The encoder of the

Table 3. Difference between transformer and modified transformer architecture

Characteristics	Transformer	Modified Transformer
Positional Encoder	Included	Does not include
Normalization layer	Layer normalization	RMS normalization
Search strategy in decoder unit	Greedy search	Beam search
Copying mechanism	Does not include	Included
Activation function in Feed forward layer	ReLU	Leaky ReLU

Figure 2. Pipeline of Explanation-to-Narrative Generation



transformer model learns the latent representations from the given input text. The decoder then generates the narratives using the learned representations from the encoder. The following subsections explain the design details of various blocks.

Record Representation

Each record can be represented as a set of {*key,value*} pairs where each feature quantile $\left(FQ_i\right)$ represents a key and the corresponding contribution $\left(FC_i\right)$ represents the value (as shown in eq. 2):

$$(\left(\forall r \in R)(\exists\left(k,v\right) \mid k = FQ_i \;\; and \;\; v = FC_i\right) \tag{2}$$

There is an exception in the representation that the last pair's key is represented by string '*reference text*' and the corresponding template annotated text represents the value. It may be noted that the value for some of the keys may be blank (or NULL) as those quantiles don't contribute towards the prediction of that instance.

Data Cleaning and Mapping

A post-hoc local explainer generates instance explanation in the form of a record $\left\{\left(Q_i, C_i\right)\right\}_{i=1}^{n}$. It discretizes each contributing feature into quantiles (Q_i) and assigns contribution values (C_i) to them. The input to the proposed model is, hence, a record of {*key,value*} pairs where key is denoted by the feature quantile $\left(Q_i\right)$ and its contribution represents the value. However, some features may drive the prediction towards the contradictory label. Hence, the authors proposed to eradicate those features before providing the record as input to the transformer model. Also, it has been observed that only few top features significantly contribute towards the prediction. Hence, as a data cleaning step, the top 'p' features are filtered out from the given set based on their contributions.

Template Generation and Data Annotation

In order to train a supervised learning-based transformer model, training data is required to teach the structure of the sentences to the model. However, as per the author's knowledge, there does not exist any such explanation-to-narrative dataset. Hence, in the paper, authors propose a template-based approach to annotate the training data. A set of human annotators are employed to generate reference phrases for each feature quantile. Multiple verbalizations are then generated for each input by combinations of the phrases to increase the diversity of text generation.

Linearization

Here, authors propose to cast the explanation-to-narrative task as a text-to-text task. Hence, they linearize the input columns into text sequences (Kale, 2020). The structured data columns are linearized into input text using a defined format, *i.e.*, "<table> <cell> v_1 <col_header> f_1 </col_header> </cell> <cell>……..</cell> <cell> v_n <col_header> f_n </col_header> </cell> </table>". The input linearized text can then be fed to the proposed transformer to generate the corresponding narrative.

Narrative Generation Model

Transformers have shown state-of-art results in text-to-text generation tasks such as machine translation, abstractive summarization, question answering, etc. The authors have employed a transformer model (Vaswani et al., 2017) which consists of an encoder and decoder to generate

the corresponding narratives. In this paper, *a modified transformer model* has been proposed for explanation-to-narrative generation.

However, there are few rare words which cannot be directly generated by the decoder module using the generative vocabulary. Hence, the authors propose to employ the copying mechanism (See et al., 2017) which copies rare words (such as the name of the features) from the input linearized text and pastes them directly to the output sequence.

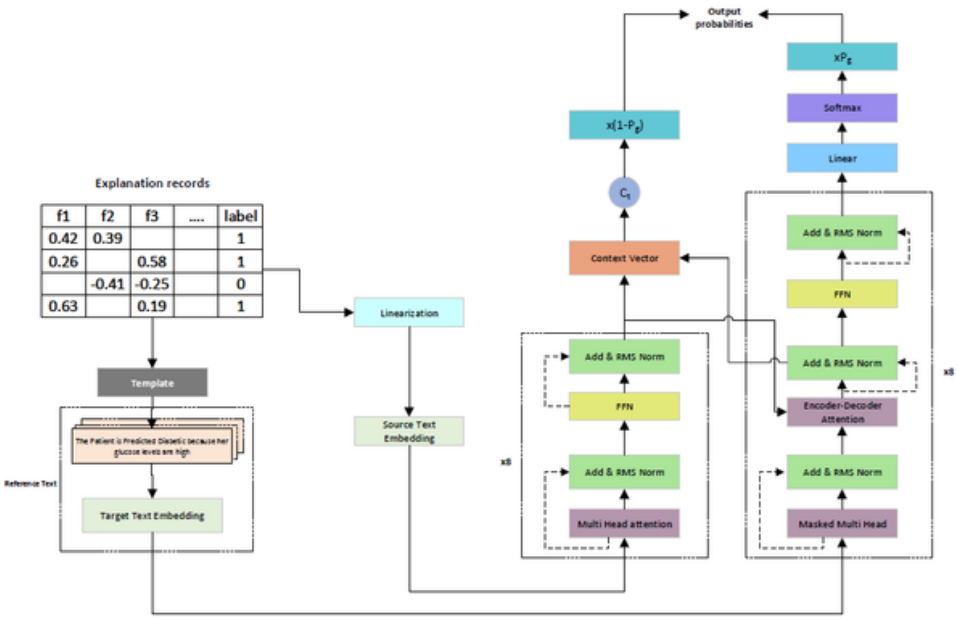
Encoder

The encoder unit of the transformer converts each discrete input symbol (including features, tags, contribution values and tag properties) into numerical representations which are then fed into the decoder part. It takes the linearized text as input. This linearized input text is initially converted to numerical vectors using input embeddings. Since, the order of the input features does not matter in the narration, hence, it has been proposed to remove the positional encoder from the vanilla transformer encoder. The main objective of the encoder is to learn better input vector representation that manages long-term dependencies and keeps semantic and syntactic properties intact.

The multi-head attention enables the model to parallelly focus on data from different dimensions. It concatenates the attention vectors of *j* heads. The resulting multi-head attention output is then fed to a feed-forward network which constitutes two linear layers and a Leaky ReLU activation layer. Each encoder in the encoder stack also consists of normalization layers which provide a residual connection around each sublayer. In this work, the authors replace the *Layer normalization layer* with the *Root Mean Square (RMS) Layer Normalization* (B. Zhang & Sennrich, 2019) as the layer normalization in standard transformer may raise computational overhead. It normalizes all the inputs to a neuron in a layer using R(c) statistic as:

$$c_i = \sum_{j=1}^{n} w_{ij} x_j$$

Figure 3. Proposed architecture of Modified transformers



$$\overline{c_i} = \frac{c_i}{R(c)} h_i$$

$$R(c) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(c_i)^2} \tag{3}$$

where w_i is the weight to the i^{th} output neuron, $\overline{c_i}$ is the normalized alternative of c_i *i.e.* the weighted summed inputs of output neurons and h_i is the rescaling gain parameter.

Decoder

The decoder part is responsible for generating narrative text to the corresponding explanations. It has the same components as the encoder with an additional encoder-decoder attention layer between the masked multi-head attention and feed forward neural network layer. The encoder-decoder attention layer of each decoder takes the input from the last encoder to learn the latent representations of the input which serves as the Key (K_{dec}), and Value (V_{dec}). Like the encoder part, the *Layer normalization layer* has been replaced with the *Root Mean Square Layer Normalization* in the decoder part as well.

In addition, the decoder layer uses greedy search to find the next word to be generated based on the highest probability. However, the greedy search technique suffers from the problem that it might miss high probability words veiled behind a low probability word. Hence, the paper uses the *beam search technique* which aids in finding an output sequence with cumulative higher probability depending on the beam size.

Copying Mechanism

Some feature attributes in the feature contribution set (FC) directly influence the text to be generated. Somehow, these cannot be directly generated by the vocabulary. Hence, it is suggested to directly copy those rare words from the attributes of the instance to be explained. The paper employs a copy mechanism while generating the output text. The copying mechanism was proposed (Gu et al., 2016) in the text-text generation to address such rare word limitations associated with Out of Vocabulary problems with text summarization. In copying mechanism, the next word w_i is generated by considering the following probabilistic distribution:

$$P_g = \sigma\left(c_t w_c^T + a_t w_a^T + x_t w_x^T + b\right) \in [0,1]$$

$$P(y_i) = P_g\left(P_{vocab}\right) + \left(1 - P_g\right) P_{enc-dec}\left(y_i\right) \tag{4}$$

where w_c, w_a, w_x and b are trainable parameters and σ is the sigmoid function. P_{vocab} is the vocabulary of the reference document. Here, P_g can be thought of as a switch to decide whether to generate word from the vocabulary or directly copy it from the input linearized text.

EXPERIMENTAL SETUP

The section presents the dataset used, various baselines, model parameters and settings, and evaluation metrics for the experiments conducted to validate the effectiveness of the proposed approach.

Dataset

The model is evaluated and validated on the structured explanations generated by a post-hoc local explainer, namely, *LIME*. The explanations are generated for the predictions of diabetes in female patients using a publicly available UCI dataset (PIMA Indians, available at: https://archive.ics.uci.edu/ml/datasets/diabetes) by a SVM model. The LIME explainer divides each feature into various quantiles to properly explain the contribution. Hence, the explanation dataset consists of feature attributes (divided into quantiles) and their corresponding contribution values for each instance. In addition, the reference text and its verbalizations are generated for each row using the proposed template approach. Figure 4 shows the explanation dataset instance and its corresponding reference texts.

Baselines

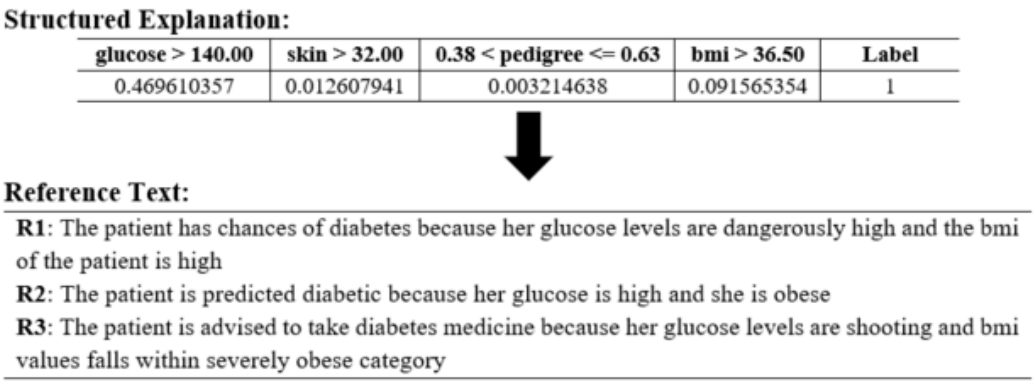
A comparison of the modified transformer has been done with the following state-of-art baseline approaches:

1. **LSTM (Hochreiter & Schmidhuber, 1997):** It is a memory-based recurrent neural network (RNN) which resolves the problem of vanishing gradient associated with the vanilla RNNs. However, it processes inputs sequentially.
2. **Vanilla Seq2Seq (Sutskever et al., 2014b):** It is an encoder-decoder model comprising of recurrent units such as RNN, LSTM or GRU as the main units. The decoder unit produces output text word-by-word conditioned on input.
3. **Vanilla Transformer (Vaswani et al., 2017):** It is an encoder-decoder model based entirely on attention mechanism. It replaces the recurrent units with the multi-head attention layers in both encoder and decoder units.
4. **Pointer-Generator Network (See et al., 2017):** It is a seq2seq model which uses copying mechanism along with coverage mechanism to keep record of the words that have been summarized for avoiding repetition.
5. **Copy-net (Gu et al., 2016):** It applies copying mechanism to the seq2seq model's decoder where the required word is copied from the input statement and pasted to the output sequence at its proper location.

Model Parameters and Settings

The modified transformer is implemented using Tensorflow library in python. It uses GPU Nvidia 1080Ti with 11 GB RAM to train the model. The vocabulary size is set to limit of 7010 for both

Figure 4. Explanation instance along with reference texts



source and target texts. The implementation uses *Adam* optimizer to speed up the process. A set of parameters, known as hyperparameters, are selected experimentally before training the model. The dataset is divided into train, test and validate sets by 80:10:10 ratios. Further, the implemented modified transformer model includes eight encoders as well as decoders units. The implemented model also uses a *learning rate scheduler* wherein the learning rate of the model is initially set to a value *i.e.* 0.01. During training, it is decreased exponentially (following eq. 4) until the validation loss stops to decrease. Let *t* be the iteration number from the total training iterations, *k* be the hyperparameter and be the learning rates, then learning rate during the training process can be calculated as:

$$\alpha = \alpha_0 * e^{-(kt)} \tag{5}$$

Also, during the testing, the beam size is set to 3.

Evaluation Metrics

The proposed model's performance is evaluated on various parameters such as BLEU, ROGUE, METEOR scores and various human evaluation parameters. Let n be the total number of candidate words, $C\left(w\right)$ be the count of each unique word *'w'* in the candidate text and $R\left(w\right)$ be the count of appearance of each *w* in the reference text.

1. **BLEU score (Papineni et al., 2002):** The weighted average of the counted number of matches between the reference and model generated (candidate) text irrespective of the words' position. It is measured on a scale of 0 to 1. Bleu score is mathematically calculated using Minimum function as:

$$Bleu = \min\left(1, e^{\left(1-\frac{m}{n}\right)}\right)\left(\prod_{j=1}^{4} p_j\right)^{\frac{1}{4}}$$

where:

$$p_j = \frac{\sum_{s \in target-vocab} \sum_{j \in s} \min\left(C\left(w\right), R\left(w\right)\right)}{n} \tag{6}$$

Table 4. Hyperparameter settings

Model	Modified Transformer
Batch Size	16
Initial learning rate	0.01
Loss function	Cross-entropy
Optimization Algorithm	Adam
Number of epochs	13
Learning rate scheduler	Exponential learning rate decay
Beam size	3
Number of Encoder layers	8
Number of Decoder layers	8

and m is the length of reference text.

2. **ROUGE (Lin, 2004):** Rouge-N generally calculates the score between the candidate text '*n-grams*' and the reference text. Mathematically, it can be defined as:

$$ROUGE-N = \frac{\sum_{r_i \in reference_{text}} \sum_{n-gram \in r_i} Count\left(n-gram, n\right)}{Num_{n_{grams}(m)}} \tag{7}$$

where Count(n-gram,n) calculates the total number of times specific n-gram appears in candidate document, Num_n_grams(m) donate the number of n-grams in reference document.

However, Rouge-N requires consecutive matching of words in an N-gram, Rouge-L matches words in a subsequence. The words can be matched in any order in the longest matching sequence using Longest Common Subsequence (LCS) algorithm.

3. **METEOR (Banerjee & Lavie, 2005):** The METEOR metric computes the score of the implemented system by aligning the candidate text to any one of the reference text. Let '*n*' is the number of mapped unigrams, then, *precision (P)* and *recall (R)* are calculated as 'n/ $l\left(c\right)$' and 'n/ $l\left(r\right)$', where $l\left(r\right)$ and $l\left(c\right)$ are reference and candidate lengths, respectively. Further, F-score can be calculated as:

$$F = \frac{10RP}{R+9P}$$

Also:

$$Penalty = \partial\left(\frac{\#\,matching\ \ chunks}{\#\,matches}\right)\ 0 \leq \partial \leq 1$$

Hence:

$$METEOR = F\left(1-Penalty\right) \tag{8}$$

It ranges between 0 and 1.

4. **Human evaluations:** Human evaluations are performed for better, thorough, and accurate comparisons. Hence, 15 human annotators were asked to score generation texts in various aspects, namely *content faithfulness, sentence fluency, relevance* and *style embodiment* on a 5-point Likert scale. Each annotator was provided with a set of information, first being sentence from the modified transformer model followed by the sentences generated from the baselines. Afterwards, they were asked to rank the information. In addition, annotators were asked to score the non-NLG and Narratives generated by the modified transformer on a 5-point Likert scale for various parameters like *Readability, Understanding* and *Trust.* For experimental study, the results on 20 test instances were evaluated and the average of all values was taken for each evaluation category.

Figure 5. 5-point Likert scale for Human evaluation

S.No.	Question	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
1.	The sentence is fluent	O	O	O	O	O
2.	The sentence is relevant to the context	O	O	O	O	O
3.	The content is readable	O	O	O	O	O
4.	The content is understandable	O	O	O	O	O
5.	It can be trusted	O	O	O	O	O
	Question		**Yes**	**No**	**Partially**	**Can't Say**
1.	It is faithful to content		O	O	O	O

RESULT AND ANALYSIS

The section presents the experimental results and analysis on the dataset for explanation-to-narrative generation. A comparative analysis of the model with above mentioned baselines on various evaluation metrics has been performed along with ablation study to visualize the effect of model's components.

Figure 6 shows the results of the LIME explainer for an instance from the dataset, followed by the feature selection and linearization outputs.

Table 5 illustrate some output examples from the proposed model and other baselines. It has been observed that the modified transformer produces texts which are closer to the human generated text (aka GOLD text). Table 6 compares the BLEU, ROUGE-1, ROUGE-2, ROUGE-3, ROUGE-L, and METEOR scores of the proposed approach with other baselines. Further, the proposed approach outperforms all the baselines for each automatic evaluation metric. Especially, the model outperforms the most commonly used LSTM baseline for machine translations by a large margin with respect to ROUGE metrics. Also, it outperforms vanilla transformer as it incorporates copying mechanism to include rare words in the output sentence. It performs better than pointer-generator and copy-net networks because unlike them, it is a transformer-based network having RMS normalization layer. Also, the proposed model shows similar results on validation dataset with 0.612 BLEU and 0.621 ROUGE evaluation scores.

Figure 6. Example instance of Pre-processed input data for the modified transformer. It includes LIME explainer output, the feature selection and linearization outputs.

glucose > 140.00	skin > 32.00	0.38 < pedigree <= 0.63	bmi > 36.50	62.00 < bp <= 72.00	1.00 < pregnant <= 3.00	Label
0.469610357	0.012607941	0.003214638	0.091565354	-0.01363433	-0.011290	1

Feature Selection

glucose > 140.00	skin > 32.00	0.38 < pedigree <= 0.63	bmi > 36.50	Label
0.469610357	0.012607941	0.003214638	0.091565354	1

Linearization

Linearized Input:

<table><cell>0.469610357<col-header>glucose > 140.00</col-header></cell><cell>0.091565354<col-header>bmi > 36.50</col-header></cell></table>

Table 5. Generated texts for linearized explanations

Linearized Input: <table><cell>0.469610357<col-header>glucose > 140.00</col-header></cell><cell>0.091565354<col-header>bmi > 36.50</col-header></cell></table>
Reference Text: The patient has chances of diabetes because her glucose levels are dangerously high and the bmi of the patient is high
LSTM: The patience have have risk diabetic because levels is shoot
Seq2Seq: The patience has diabetes because glucogen level high and or bp is obese bp
Vanilla Transformer: The patient diabetic reason glucose level are high and bmi heavy obese bp
Pointer Generator: The patient have diabetic because glucose levels is high bmi obese overweight
Modified Transformer: The patient has chances of diabetes because blood tests report high glucose and patient' s bmi values falls within severly obese category

Table 6. Automatic evaluation results

Model	BLEU	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-L	METEOR
LSTM	0.561	0.465	0.468	0.472	0.481	0.416
Seq2Seq	0.564	0.469	0.471	0.475	0.485	0.420
Transformer	0.586	0.512	0.522	0.527	0.562	0.469
Pointer-Generator	0.592	0.481	0.493	0.501	0.518	0.481
Copy-net	0.598	0.492	0.519	0.505	0.521	0.483
Modified transformer	**0.639**	**0.651**	**0.665**	**0.674**	**0.678**	**0.569**

In addition, table 7 provides the comparative analysis of the approaches for human evaluation categories. The human annotators were asked to score the generated texts on a Likert scale which consists of value points as: strongly disagree (1), disagree (2), neither agree or disagree (3), agree (4) and strongly agree (5). The scores from all the annotators for each evaluation category is averaged to find the human evaluation score. It has been observed that modified transformer ($p<0.001$) model performs better than other baselines *w.r.t. sentence fluency* and *relevance*. The inclusion of copying mechanism along with beam search strategy helped the model outperform the baseline techniques. However, Pointer generator network performs marginally better than the model *w.r.t. style embodiment* due to coverage mechanism. Further, table 8 shows the responses of human annotators for content faithfulness in terms of yes, no, partially or can't. The annotators agreed that more than 79% of the text generated by the modified transformer model is faithful to the content. However, 5% of the generated text is partially correct and 6% is incorrect. Rest, many annotators couldn't tell anything about the content generated. The results suggest that the modified model generates more factually correct statements compared to all baseline models.

In addition to this, the authors compared the model with non-NLG output *i.e.* the structured output from the explainer (as shown in table 9). On an average, the human annotators understood the narratives (NLG output) better than the non-NLG output. Also, generated narratives were easy to

Table 7. Human evaluation results

Evaluation category	LSTM	Seq2Seq	Transformer	Pointer-Generator	Copy-net	Modified Transformer
Sentence Fluency	-0.56	-0.32	0.34	0.32	0.36	**0.79**
Relevance	-0.21	-0.05	0.45	0.48	0.63	**0.86**
Style embodiment	0.25	0.32	0.36	**0.82**	0.45	0.76

Table 8. Responses of content faithfulness

Response	LSTM	Seq2Seq	Transformer	Pointer-Generator	Copy-net	Modified Transformer
Yes	30%	42%	76%	69%	71%	82%
No	52%	34%	14%	20%	19%	6%
Partially	08%	21%	7%	6%	7%	5%
Can't decide	10%	3%	3%	5%	3%	7%

Table 9. Comparison of narrative with non-NLG output (-1 to 1)

Parameter	Non-NLG	Narrative
Understandability	-0.72	**0.78**
Readability	0.2	**0.89**
Trust	-0.45	**0.84**

read and built trust as humans understand text better than facts. The proposed model proposes a modified transformer with encoder-decoder self-attention layers. Hence, it can be observed that the training complexity associated with the approach is $O\left(t^2\right)$ where t is the sequence length.

Ablation Study

This section focusses on conducting extensive ablation studies to understand contribution of each component of the model. Table 10 shows the results for different evaluation metrics.

Here, ✓ means the inclusion of corresponding column component and × means removal of the column component. Here, the paper uses "CM" for copy mechanism, "RMS" for Root Mean Square Normalization, "BS" for Beam Search and "LRELU" for Leaky ReLU.

Effect of Copying Mechanism

From the table 8, it is evident that copying mechanism plays an important role in increasing the efficiency and faithfulness of the text. Without any component, the model acts as a vanilla transformer. Incorporating copying mechanism in vanilla transformer for explanation-to-narrative generation helps in increasing the scores in all the automatic metrics as it directly copies the rare words from the input to the narrative text.

Table 10. Ablation study

CM	RMS	BS	LReLU	BLEU	ROUGE	METEOR	Time taken (s)
×	×	×	×	0.42	0.48	0.412	0.79
✓	×	×	✓	0.62	0.656	0.51	0.72
✓	✓	×	×	0.626	0.668	0.528	0.61
✓	✓	✓	×	0.631	0.671	0.559	0.59
✓	✓	✓	✓	0.639	0.678	0.569	0.32

Effect of RMS Normalization and Leaky ReLU

In this part, the authors compared modified transformer model with the transformer model with copying mechanism. It is quite evident from the table that with the inclusion of RMS normalization in the encoder and decoder part of the transformer, the speed of the network increases considerably by 28% while maintaining comparable performance. Also, with the replacement of ReLU layer with the Leaky ReLU, the time taken for training the network decreases by 15% while maintaining performance.

Effect of Beam Search

The model is also tested for the Beam Search component. When beam search is added in the decoder unit, each of the automatic metrics are improved slightly as compared to vanilla transformer. This is, however, reasonable because the beam search aims at increasing output quality and faithfulness.

CONCLUSION

The explanations generated by the state-of-art post-hoc XAI techniques are mainly in the form of structured data. As a non-expert user finds it difficult to interpret as well as trust the explanations, the need to generate narratives from such explanations necessitates the extensive use of NLG techniques. The paper proposed an approach for automated generation of narratives using a modified transformer with copying mechanism for data-to-text generation tasks. Initially, the input data is linearized to convert the task into text-to-text. Further, a verbalizational template-based approach is also proposed to annotate the data for training and testing the transformer. Our modified transformer uses the RMS normalization layer in the encoder and decoder along with a copying mechanism that includes rare words from input to output sentence. The proposed approach generated high quality narratives for structured explanations and outperformed various baselines for data-to-text on automatic and manual evaluations with a BLEU score of 0.639. During the process of comparing the narratives with non-NLG output of the explainer, the human annotators preferred the narratives by a significant margin. Also, the ablation study exhibited that copying mechanism plays an important role in increasing the efficiency and faithfulness of the text and the inclusion of RMS normalization layer, Leaky ReLU and Beam search enhance the performance of the model.

REFERENCES

Alonso, J. M., & Bugarin, A. (2019). ExpliClas: Automatic Generation of Explanations in Natural Language for Weka Classifiers. *2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 1-6. doi:10.1109/FUZZ-IEEE.2019.8859018

Alonso, J. M., Barro, S., Bugarin, A., van Deemter, K., Gardent, C., Gatt, A., Reiter, E., Sierra, C., Theune, M., Tintarev, N., Yano, H., & Budzynska, K. (2020). Interactive Natural Language Technology for Explainable Artificial Intelligence. *1st Workshop on Foundations of Trustworthy AI Integrating Learning, Optimisation and Reasoning (TAILOR), at the European Conference on Artificial Intelligence (ECAI)*.

Bahdanau, D., Cho, K., & Bengio, Y. (2014). *Neural Machine Translation by Jointly Learning to Align and Translate*. ArXiv Preprint.

Banerjee, S., & Lavie, A. (2005). METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. *Proceedings of the Acl Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, 65–72.

Belz, A., & Kow, E. (2009). System building cost vs. output quality in data-to-text generation. *Proceedings of the 12th European Workshop on Natural Language Generation (ENLG 2009)*, 16-24. doi:10.3115/1610195.1610198

Cheng, Y., Zhang, X., Wang, X., Zhao, H., Yu, Y., Wang, X., & de Pablos, P. O. (2021). Rethinking the Development of Technology-Enhanced Learning and the Role of Cognitive Computing. *International Journal on Semantic Web and Information Systems*, *17*(1), 67–96. doi:10.4018/IJSWIS.2021010104

Ehsan, U., Harrison, B., Chan, L., & Riedl, M. O. (2018). Rationalization: A neural machine translation approach to generating natural language explanations. *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, 81–87. doi:10.1145/3278721.3278736

EU. (2019). *Communication Artificial Intelligence for Europe*. https://ec.europa.eu/digital-single-market/en/news/communication-artificial-intelligence-europe

Fayoumi, A. G., & Hajjar, A. F. (2020). Advanced learning analytics in academic education: Academic performance forecasting based on an artificial neural network. *International Journal on Semantic Web and Information Systems*, *16*(3), 70–87. doi:10.4018/IJSWIS.2020070105

Fiorini, R. A. (2020). Computational intelligence from autonomous system to super-smart society and beyond. *International Journal of Software Science and Computational Intelligence*, *12*(3), 1–13. doi:10.4018/IJSSCI.2020070101

Forrest, J., Sripada, S., Pang, W., & Coghill, G. (2018). Towards making NLG a voice for interpretable Machine Learning. *Proceedings of The 11th International Natural Language Generation Conference*. doi:10.18653/v1/W18-6522

Gardent, C., Shimorina, A., Narayan, S., & Perez-Beltrachini, L. (2017). The WebNLG challenge: Generating text from RDF data. *Proceedings of the 10th International Conference on Natural Language Generation*, 124–133. doi:10.18653/v1/W17-3518

Gatt, A., & Reiter, E. (2009). SimpleNLG: A realisation engine for practical applications. *Proceedings of the 12th European Workshop on Natural Language Generation (ENLG 2009)*, 90–93. doi:10.3115/1610195.1610208

Goldberg, E., Driedger, N., & Kittredge, R. I. (1994). Using natural-language processing to produce weather forecasts. *IEEE Intelligent Systems*, *9*(2), 45–53.

Gu, J., Lu, Z., Li, H., & Li, V. O. (2016). *Incorporating copying mechanism in sequence-to-sequence learning*. doi:10.18653/v1/P16-1154

Gunning, D., & Aha, D. (2019). DARPA's Explainable Artificial Intelligence (XAI) Program. *AI Magazine*, *40*(2), 44–58. Advance online publication. doi:10.1609/aimag.v40i2.2850

Gupta, B. B., Li, K.-C., Leung, V. C., Psannis, K. E., & Yamaguchi, S. (2021). Blockchain-assisted secure fine-grained searchable encryption for a cloud-based healthcare cyber-physical system. *IEEE/CAA Journal of Automatica Sinica*.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780. doi:10.1162/neco.1997.9.8.1735 PMID:9377276

Inan, E., & Dikenelli, O. (2021). A Semantic-Embedding Model-Driven Seq2Seq Method for Domain-Oriented Entity Linking on Resource-Restricted Devices. *International Journal on Semantic Web and Information Systems*, *17*(3), 73–87. doi:10.4018/IJSWIS.2021070105

Kale, M. (2020). *Text-to-text pre-training for data-to-text tasks*. ArXiv Preprint.

Langner, B., Vogel, S., & Black, A. W. (2010). Evaluating a dialog language generation system: Comparing the MOUNTAIN system to other NLG approaches. *Eleventh Annual Conference of the International Speech Communication Association*. doi:10.21437/Interspeech.2010-353

Lebret, R., Grangier, D., & Auli, M. (2016). *Neural text generation from structured data with application to the biography domain*. doi:10.18653/v1/D16-1128

Li, J., Tang, T., Zhao, W. X., & Wen, J.-R. (2021). *Pretrained Language Models for Text Generation: A Survey.* ArXiv Preprint.

Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*, 74-81.

Moosavi, N. S., Ruckle, A., Roth, D., & Gurevych, I. (2021). *Learning to Reason for Text Generation from Scientific Tables.* ArXiv Preprint.

Nie, F., & Wang, J., & Pan, C.-Y. (2019). An Encoder with non-Sequential Dependency for Neural Data-to-Text Generation. *Proceedings of the 12th International Conference on Natural Language Generation*, 141–146. doi:10.18653/v1/W19-8619

Novikova, J., Duvsek, O., & Rieser, V. (2017). *The E2E dataset: New challenges for end-to-end generation.* ArXiv Preprint.

Pandey, A., & Banerjee, S. (2019). Test suite optimization using firefly and genetic algorithm. *International Journal of Software Science and Computational Intelligence*, *11*(1), 31–46. doi:10.4018/IJSSCI.2019010103

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 311–318.

Park, D. H., Hendricks, L. A., Akata, Z., Rohrbach, A., Schiele, B., Darrell, T., & Rohrbach, M. (2018). Multimodal explanations: Justifying decisions and pointing to the evidence. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8779–8788. doi:10.1109/CVPR.2018.00915

Pereira, J. C., Teixeira, A., & Pinto, J. S. (2015). Towards a hybrid nlg system for data2text in portuguese. *2015 10th Iberian Conference on Information Systems and Technologies (CISTI)*, 1-6.

Reiter, E., & Dale, R. (1997). Building applied natural language generation systems. *Natural Language Engineering*, *3*(1), 57–87. doi:10.1017/S1351324997001502

Reiter, E., Sripada, S., Hunter, J., Yu, J., & Davy, I. (2005). Choosing words in computer-generated weather forecasts. *Artificial Intelligence*, *167*(1-2), 137–169. doi:10.1016/j.artint.2005.06.006

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should i trust you?" Explaining the predictions of any classifier. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. doi:10.1145/2939672.2939778

Roth, A. E. (1988). A value for n-person games. The Shapley Value.

Sarivougioukas, J., & Vagelatos, A. (2020). Modeling deep learning neural networks with denotational mathematics in UbiHealth environment. *International Journal of Software Science and Computational Intelligence*, *12*(3), 14–27. doi:10.4018/IJSSCI.2020070102

See, A., Liu, P. J., & Manning, C. D. (2017). *Get to the point: Summarization with pointer-generator networks*. ArXiv Preprint.

Singh, S. K., & Sachan, M. K. (2021). Classification of Code-Mixed Bilingual Phonetic Text Using Sentiment Analysis. *International Journal on Semantic Web and Information Systems*, *17*(2), 59–78. doi:10.4018/IJSWIS.2021040104

Sun, Y., Gu, F., & Ji, S. (2019). Sparse reconstruction of piezoelectric signal for phased array structural health monitoring. *International Journal of High Performance Computing and Networking*, *14*(4), 466–472. doi:10.1504/IJHPCN.2019.102354

Sutskever, I., Vinyals, O., & Le, Q. V. (2014a). *Sequence to sequence learning with neural networks*. ArXiv Preprint.

Sutskever, I., Vinyals, O., & Le, Q. V. (2014b). Sequence to sequence learning with neural networks. *ArXiv Preprint, 1409*(3215).

Tolstikhin, I., Bousquet, O., Gelly, S., & Schoelkopf, B. (2017). *Wasserstein auto-encoders*. ArXiv Preprint.

van der Lee, C., Krahmer, E., & Wubben, S. (2017). PASS: A Dutch data-to-text system for soccer, targeted towards specific audiences. *Proceedings of the 10th International Conference on Natural Language Generation*, 95-104. doi:10.18653/v1/W17-3513

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). *Attention is all you need*. ArXiv Preprint.

Wiseman, S., Shieber, S. M., & Rush, A. M. (2017). *Challenges in data-to-document generation*. ArXiv Preprint. doi:10.18653/v1/D17-1239

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., & Bengio, Y. (2015). Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. *International Conference on Machine Learning*, 2048–2057.

Xu, X., He, T., & Wang, H. (2020). A Novel Data-to-Text Generation Model with Transformer Planning and a Wasserstein Auto-Encoder. *2020 IEEE International Conference on Services Computing (SCC)*, 337–344. doi:10.1109/SCC49832.2020.00051

Yermakov, R., Drago, N., & Ziletti, A. (2021). *Biomedical Data-to-Text Generation via Fine-Tuning Transformers*. ArXiv Preprint.

Zhang, B., & Sennrich, R. (2019). *Root Mean Square Layer Normalization*. ArXiv Preprint.

Zhang, Z., Chen, P., McGough, M., Xing, F., Wang, C., Bui, M., Xie, Y., Sapkota, M., Cui, L., Dhillon, J., Ahmad, N., Khalil, F. K., Dickinson, S. I., Shi, X., Liu, F., Su, H., Cai, J., & Yang, L. (2019). Pathologist-level interpretable whole-slide cancer diagnosis with deep learning. *Nature Machine Intelligence*, *1*(5), 236–245. doi:10.1038/s42256-019-0052-1

Diksha Malhotra received her Master's degree in Computer Science Engineering from Punjab Engineering College (Deemed to be University), India in 2018. Currently, she is Research Scholar (pursuing Ph.D.) in the Department of Computer Science Engineering from Punjab Engineering College (Deemed to be University), India. Her research interests include Natural Language Processing, Explainable Artificial Intelligence, Distributed Computing and Blockchain, Databases, and Information Retrieval.

Poonam Saini received her Ph.D. degree in Computer Engineering from National Institute of Technology, Kurukshetra, India in 2013 and M.Tech from UIET, Kurukshetra University, Kurukshetra, India in 2006. She has received B. Tech. in Information Technology from Kurukshetra University, Kurukshetra, India in 2003. She is currently working as Assistant Professor in Computer Science and Engineering at PEC University of Technology (formerly Punjab Engineering College), Chandigarh, India. Her research interest includes Fault-Tolerant Distributed Computing Systems, Mobile Computing, Ad hoc Networks, Wireless Sensors Networks, Cloud Computing and Security, Blockchain and Artificial Intelligence.

Awadhesh Kumar Singh is currently Professor, Computer Engineering Department, National Institute of Technology, Kurukshetra, India [2013 to date]. His research interest includes Fault-Tolerant Distributed Computing Systems, Mobile Computing, Ad hoc Networks, Cloud Computing and Security. He has a teaching and research experience of more than 25 years. He has published more than 150 publications in reputed journals and conferences. He has supervised more than 30 dissertations at M.Tech. and PhD level.