Natural Language Explanations for Machine Learning Classification Decisions

James Burton
Department of Computer Science
Durham University
Durham, UK
james.burton@durham.ac.uk

Noura Al Moubayed
Department of Computer Science
Durham University
Durham, UK
noura.al-moubayed@durham.ac.uk

Amir Enshaei
Faculty of Medical Sciences
Newcastle University
Newcastle, UK
amir.enshaei@newcastle.ac.uk

***Abstract*—This paper addresses the challenge of providing understandable explanations for machine learning classification decisions. To do this, we introduce a dataset of expert-written textual explanations paired with numerical explanations, forming a data-to-text generation task. We fine-tune BART and T5 language models on this dataset to generate natural language explanations by linearizing the information represented by explainable output graphs. We find that the models can produce fluent and largely accurate textual explanations. We experiment with various configurations and see that an augmented dataset leads to a reduced error rate. Additionally, we probe the numerical explanations more directly by fine-tuning BART and T5 on a question-answer task and achieved an accuracy of 91% with T5.**

***Index Terms*—explainability, data-to-text, natural language generation**

I. Introduction

Despite the advantages of using machine learning (ML) techniques to solve complex problems, there are particular application areas, such as finance, health, and criminal justice, that have been hesitant to adopt ML approaches [1]–[3] with stakeholders concerned about the consequences a wrong decision could have. For these areas in particular, the notion of explainability is critical. Stakeholders not only want to know *what* the model is predicting but also *why*. Understanding the factors influencing an ML model's prediction enables actionable business choices, transparency, and confidence. Furthermore, this aligns them with the recently proposed EU Artificial Intelligence Act [4].

Simple predictive algorithms such as linear models, generalized additive models, and shallow decision trees are inherently explainable since they are easy to understand, and sourcing the reason for classification output decisions is simple [3], [5]. However, for complicated architectures such as deep neural networks, it is challenging to trace which features were relied upon most for making the decision [6], [7] as these black-box models employ billions of parameters to make predictions, making them difficult to troubleshoot and trust.

In recent years, there has been an effort to increase transparency in the decision-making process of black-box models used for predictions and incorporate eXplainable AI (XAI) techniques. In a typical explainability pipeline, as shown in Fig. 1 (left), a trained classifier will make a prediction. To make a local-level explanation, the XAI technique will utilize the prediction and the classifier to yield importance scores for each input feature.

Four commonly used eXplainable AI (XAI) techniques include Local Interpretable Model-Agnostic Explanations (LIME) [8], SHapley Additive exPlanations (SHAP) [9], Integrated Gradients (IG) [10], and Layer-wise Relevance Propagation (LRP) [11]. Although distinct, these techniques all produce feature importance values that quantify the contribution of each feature to the prediction.

Graphs and figures are commonly used to communicate the contributions of each variable used to arrive at a given prediction. For example, Fig. 3 shows a graph generated using LIME to explain why a given "wine" was labeled as "high quality". These graphs produced by XAI techniques indicate which features are positive (supporting the prediction output), negative (contradicting the prediction output), and neutral (having a negligible influence on the prediction decision). However, for non-experts, it can be challenging to fully understand these figures.

Large, pre-trained language models are trained on a vast text corpus, giving them a broad generalized understanding of language. Fine-tuning these models for specific tasks has been shown to improve their task-specific understanding, even with limited training data [12]–[14]. Two such language models are T5 [15], and BART [16]. T5 is a multitask-trained transformer model trained on several unsupervised and supervised NLP tasks, such as classification, summarization, and translation. BART [16] is a transformer-based denoising autoencoder trained to reconstruct the original text from a corrupted input.

In this paper, we propose a new task: given a classifier prediction and a subsequent local-level explanation, produce a narrative that describes the explanation. The narrative should be fluent and factually accurate to provide clarity to the end user when provided alongside a figure. The task is designed to be ambivalent to the choice of explainability technique. The only requirement is that the classifier produces a probability estimation across the classes and that the XAI technique produces a score for each input feature.

To achieve this task, we consulted computer science experts with knowledge of explainability to create a new dataset: TEXtual Explanation Narratives (TEXEN). TEXEN comprises local-level explainability outputs and written narratives that

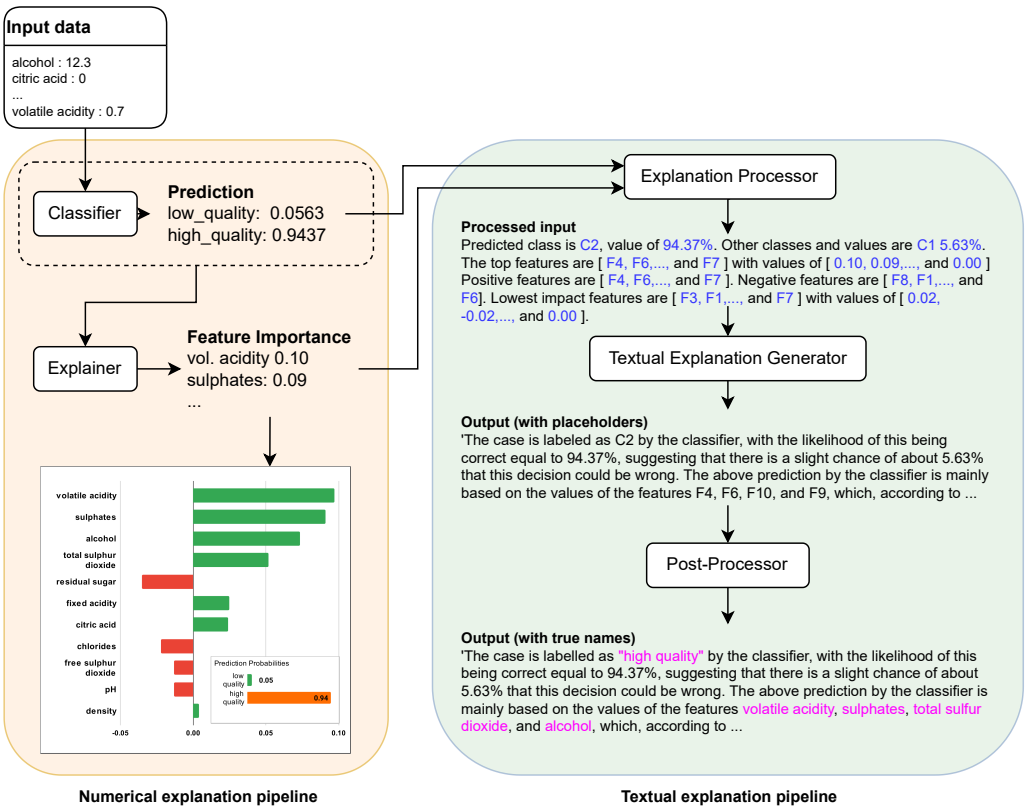


Fig. 1. A typical numerical explanation pipeline is on the left. The proposed, complimentary textual explanation pipeline is on the right. The predicted probabilities and feature importance values are processed into a template, values in blue. The Textual Explanation Generator is trained with placeholders, the Post-Processor replaces placeholders for actual names (in pink).

explain in plain text what the numerical explanations are showing. After sifting for quality and factual accuracy, TEXEN contains 496 explanation-text pairs.

In this paper, we fine-tune T5 and BART on this dataset to generate automatic textual explanations. We also train on an augmented version of TEXEN, using the same narratives but reshuffled feature names to artificially increase the size of the training set. Additionally, as a comparison, we train a question-answer model to respond to questions more directly. Our contributions through this work include:

- Introducing a new dataset[1] for generating textual explanations of a given classification decision. The textual explanations are written by computer science experts and checked manually to ensure that they appropriately reflect the contribution of input features, as produced by numerical explainability methods. To the best of our knowledge, this study is the first of its kind to focus on generating textual explanations via neural NLG.
- Experimentation and evaluation with state-of-the-art neural pre-trained language models demonstrate the opportunities and challenges for future research on this structured data-to-text generation task.

[1]https://github.com/jameswburton18/LocalLevelExplanations

II. Related Works

Structured data-to-text generation is an NLG task where descriptive texts are generated in natural language, verbalizing the information from source data such as graphs and tables [17]–[19]. The ML algorithms and techniques employed for generating text from structured data can be classified into two main groups: pipelined and end-to-end techniques [20]–[22]. Earlier NLG works predominantly employed pipeline-based techniques where the text generation process was divided into different stages: content determination, text planning, sentence planning, and surface realization modules [23], [24]. At the heart of pipeline techniques are the linguistic rules and heuristics used to select and populate pre-defined templates and schemas [17], [23], [25]. Pipeline techniques are defined within a fixed structure, so although more straightforward, they are less flexible and produce less diverse outputs than generative techniques.

In recent years, end-to-end data-to-text generation has gained a lot of attention, and this growing interest is driven by recent advancements in deep neural networks [19], [20], [26]. Another appeal of neural NLG is that texts are generated automatically from the data without needing handcrafted rules. Applications of deep neural NLG approaches include table-to-text [13], [14], [19], [27], table-based question answering [28]–[30], and graph-to-text generation [31], [32].

A significant challenge of deep neural approaches is that they require a large amount of clean data to achieve higher generation performance. Recent works [12], [13], [26], [33] indicate that utilizing pre-trained language models such as GPT [34], BERT [35], BART [16], and T5 [15] can further improve text generation performance when solving NLG tasks with a limited amount of data. Since these language models are trained with text-to-text generation objectives, applications to data-to-text require converting the structured data into flat-string (linearization).

Our work is in line with previous work by [36], [37]. They developed ExpliClas, a rule-based NLG system for generating multimodal (graphical and textual) explanations for classifiers implemented with WEKA [38]. Unlike [36], [37], our textual explanations are generated end-to-end with neural NLG. Furthermore, the trained neural NLG models can generate textual explanations based on the graphical visualization produced by any arbitrary XAI technique. To the best of our knowledge, there are no existing works exploring the application of neural NLG for generating textual explanations describing the intuition behind classification decisions.

III. Local-level Textual Explanations Dataset

A. Textual Explanation Narratives Dataset

TEXEN consists of pairs of explanations: one output of a local-level explanation method, which we refer to as a numerical explanation (an example is shown in Fig. 2), and one textual narrative, which describes in plain text what the numerical explanation is showing (such as in Fig. 4).

First, to collect the numerical explanations, we trained a selection of models on a selection of tasks. Ten different model types were used, including Support Vector Machines, Logistic Regression, Deep Neural Networks, and Random Forests. Using random samples from the test sets, local-level explanations were generated using four explainable AI techniques: LIME, SHAP, IG, and LRP. These techniques generated numerical scores for each input feature, indicating their relative influence on the classification decision. However, it is necessary to reiterate that these scores do not reflect the accuracy of the classifier but rather provide insight into which features were most important in the decision-making process. Statistics on how the numerical explanations were collected are in Table I.

To collect narratives, eight computer science experts were shown a chart (as in Fig. 3) and asked to summarize it in a single text box. These narratives are intended to describe the prediction as a whole. However, in order to guide the annotators, they were asked to provide textual explanations that answered the following questions:

i Summarize the prediction made for the test case under consideration along with the likelihood of the different possible class labels.

ii Summarize the top features influencing the model's decision.

iii Summarize the features with moderate to low influence on the model's decision.

Predicted Label	high quality
Prediction Probabilities	low quality: 5.63%
	high quality: 94.37%
Attributions	
Feature Name	**Importance Value**
volatile acidity	0.10
sulphates	0.09
alcohol	0.07
total sulphur dioxide	0.05
residual sugar	-0.04
fixed acidity	0.02
citric acid	0.02
chlorides	-0.02
free sulphur dioxide	-0.01
pH	-0.01
density	0.00

Fig. 2. An example of a numerical explanation

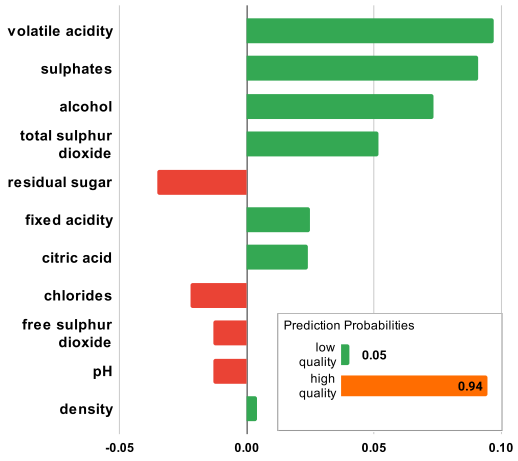


Fig. 3. An example output graph from LIME, corresponding to Fig. 2

The wine is labelled as "high quality" by the classifier, with the likelihood of this being correct equal to 94.37%, suggesting that there is a slight chance of about 5.63% that this decision could be wrong. The above prediction by the classifier is mainly based on the values of the features volatile acidity, sulphates, total sulfur dioxide, and alcohol, which, according to the analysis performed, offer strong positive support for the prediction. The other variables with a positive influence on the decision are citric acid, fixed acidity, and density, further cementing the belief in the decision made here. The 5.63% likelihood of the "low quality" can be blamed on the negative influence of chlorides, residual sugar, free sulfur dioxide, and pH, decreasing the likelihood of the "high quality" label assigned to the wine under consideration. In summary, the confidence level of 94.37% in the "high quality" label assignment is mainly due to the strong positive influence of sulphates, volatile acidity, and alcohol.

Fig. 4. An example of a textual explanation corresponding to Fig. 2

TABLE I
STATISTICS ON DATA USED FOR NUMERICAL EXPLANATION GENERATION

Property	Value
Datasets used:	40
Models used	10
Records per dataset: Mean / S.D.	11.7 / 3.4
Records per model: Mean / S.D.	42.7 / 34.3
Input features per record: Mean / S.D.	18.7 / 15.0

iv Compare the features with positive contributions to those with negative contributions resulting in the classification decision.

We collected 700 textual explanations from the experts, which were manually checked to ensure they correctly articulated the information in the corresponding explanation graph. A majority (469) were shown to accurately capture the information and correctly answer the questions posed to the annotators. Feature and class names were substituted for placeholders and randomized to prevent train-test leakage. The data was divided randomly into training, validation, and test sets (328/47/94). Statistics about the dataset introduced are summarized in Table II.

B. TEXEN-Augmented

We hypothesized that our limited training set might impede model performance. We proposed a new augmented training set constructed from the original numerical explanation narrative pairs and substituting it in a newly randomized set of feature and class name placeholders. For each item in the training set of TEXEN, the feature and class names were re-randomized ten times so that the augmented dataset contains 3421 records (train/validation/test split: 3280/47/94). Validation and test sets do not undergo this augmentation process such that the direct comparison between models can occur.

Aside from the feature and class placeholders, the narratives will remain identical; the work does not attempt to rewrite the narratives in a new way. In re-randomizing placeholders, we loosen the dependency on learning spurious correlations and encourage the model to learn the link between the features and feature values in the input and the features mentioned in the text.

IV. TEXTUAL EXPLANATION GENERATION

The purpose of the textual explanation generation task is to add to an existing numerical explanation pipeline to clarify a local numerical explanation shown as a graph. Here we outline a typical numerical explanation pipeline (Fig. 1, left), formally define the problem statement, and subsequently detail the proposed textual explanation pipeline (Fig. 1, right).

A. Numerical explanation pipeline

A typical numerical explanation pipeline is shown on the left of Fig. 1. A numerical explanation pipeline aims to explain why a classifier made the decision it did for a particular input data record. In this work, we only consider explanation methods that produce feature importance scores that can then be shown as a graph. A numerical explanation pipeline consists of the following:

1) Classifier: Given a test case, a trained classifier generates the classification output decision. This *prediction* is in the form of class labels and their respective predicted probabilities.

2) Explainer: : The explainer's task is to generate *feature importance scores* for each input feature which explains the classification decision of the particular test case, given a trained classifier and the prediction decision. These feature importance scores are then represented as a *graph*, the final output of a typical local-level explanation. In this paper, the explainability techniques used are LIME, SHAP, IG, and LRP.

TABLE II
STATISTICS FOR THE LOCAL-LEVEL TEXTUAL EXPLANATION DATASET

Property	Value
Size: Train / Validation / Test	328 / 47 / 94
Words per narrative: Mean / S.D.	188 / 47
Unique words	2466

B. Problem Definition

Given a numerical explanation, the task is to produce a narrative that explains in text what the graph is showing. Formally, a numerical explanation consists of the following: m class names $\mathbf{c} = [c_1, ..., c_m]$, their associated class probabilities $\mathbf{p} = [p_1, ..., p_m]$, n feature names $\mathbf{f} = [f_1, ..., f_n]$ and their associated feature importance values $\mathbf{v} = [v_1, ..., v_n]$. The n^+ features with values $v_i \geq 0$ and the n^- features with values $v_j < 0$ such that $n^+ + n^- = n$ are formally defined as $\mathbf{f}^+ = \left[f_i^+, ..., f_{n^+}^+\right]$ and $\mathbf{f}^- = \left[f_j^-, ..., f_{n^-}^-\right]$, respectively, where $\mathbf{f}^+$ and $\mathbf{f}^-$ are subsets of $\mathbf{f}$, such that $\mathbf{f} = \mathbf{f}^+ \cup \mathbf{f}^-$.

C. Textual Explanation pipeline

The proposed textual explanation pipeline has three components:

- An Explanation Processor that converts a numerical explanation into an input string with placeholder features and class names
- A language model trained for text-to-text generation
- A post-processor to replace the placeholders with actual feature and class names.

In this work, we have class names, probabilities, feature names and feature importance values for each input. Therefore, in order to use text-based language models, we must format this structured data into an appropriate string template.

1) Explanation Processor: The input to the explanation processor is a numerical explanation, as defined above. At this stage $\mathbf{c}$, $\mathbf{p}$, $\mathbf{f}$ and $\mathbf{v}$ are reordered from highest absolute value to lowest to match the presentation of the output graphs. A set of class name placeholders $C1$, ..., Cm is shuffled and substituted in for each item in $\mathbf{c}$. We repeat this approach for feature names, where each feature name in $\mathbf{f}$ is substituted for

a placeholder in the shuffled set of $F1$, ..., Fn. Substitution is done so the model can transfer its learning from task to task; furthermore, tokenized inputs will not have to be truncated due to long feature names. This step is also crucial to prevent the model from learning from tasks it has seen before.

Following [14], [33], [39], the final stage of the "Explainer Processor" involves linearization of the data into a flat string: $\mathbf{p}$, $\mathbf{v}$ and the newly substituted $\mathbf{c}$ and $\mathbf{f}$ are formatted into the template below. A cap, top_n, set at $min(n, 10)$ or $min(n, 20)$ during training, is used to limit the number of top features passed into the model and positive and negative features are subsets of the capped top features, such that $top\_n^+ + top\_n^- = top\_n$; the lowest impact features are not affected. Note that the top features and values are formatted so that only the final value is preceded by "and".

> Predicted class is $<c_1>$, value of $<p_1>$. Other classes and values are $<c_2>$ $<p_2>$ & ... & $<c_m>$ $<p_m>$. Top features are [$<f_1>$, ..., and $<f_{top\_n}>$], with values [$<v_1>$, ..., and $<v_{top\_n}>$]. Positive features are [$<f_i^+>$, ..., and $<f_{top\_n^+}^+>$]. Negative features are [$<f_j^->$, ..., and $f_{top\_n^-}^->$]. Lowest impact features are [$<f_{n-4}>$, ..., and $<f_n>$] with values [$<v_{n-4}>$, ..., and $<v_n>$].

2) Textual Explanation Generator: The tokenized inputs are passed into a pre-trained language model. We experiment with both T5 [15] and BART [16]. These language models are trained in a sequence-to-sequence fashion, using the collected textual explanations (with placeholders substituted in) as reference texts. In training, this is the final stage. In testing, the output (with placeholders) is passed to the *Post-Processor*.

3) Post-Processor: The function of the post-processor is simply to reverse the placeholder substitution process. Using regular expressions, class and feature name placeholders are identified and mapped back to the original string values. This stage is not active during training when the model requires a consistent way of representing the data, but only during inference when it is helpful to report the true names.

V. Question-Answer task

Here we investigate question-answering using synthetically generated numerical explanations by assigning random feature attributions and class values to class and feature placeholders. A training dataset of 27,000 records and a validation dataset of 3,000 records are generated in this manner. The question-answer pairs are created by randomly selecting a question from a pool of 8 templates for each numerical explanation. The test set consists of 469 records, using numerical explanations from the TEXEN train, validation, and test sets combined. For the test set, one question-answer pair is generated per numerical explanation.

Numerical explanations are synthetically generated in the following manner: Classes $C1$ and $C2$ have a random percentage probability (0.00%-100.00%) assigned to them, such that probabilities $p1+p2 = 1$. top_n is set as a random number between 6-20, and then each of which is given a random feature placeholder and a random feature attribution between -0.50 and 0.50.

> Predicted class is $<c_1>$, value of $<p_1>$. Other classes and values are $<c_2>$ $<p_2>$ & ... & $<c_m>$ $<p_m>$. Top features are [$<f_1>$, ..., and $<f_{top\_n}>$], with values [$<v_1$, ..., and $<v_{top\_n}>$]. Positive features are [$<f_i^+>$, ..., and $<f_{top\_n^+}^+>$]. Negative features are [$<f_j^->$, ..., and $f_{top\_n^-}^->$]. Lowest impact features are [$<f_{n-4}>$, ..., and $<f_n>$] with values [$<v_{n-4}>$, ..., and $<v_n>$]. Answer the following question: $<Q>$

The input string (above) is in the same format as in the textual explanation generation task but with an additional prompt and subsequent question, Q, which is selected at random from the eight question templates below:

Questions:

1) **What is the prediction for class *X*?** Class *X* is randomly chosen. The required answer is the predicted class probability for class *X*.
2) **What is the value of *X*?** *X* is a random feature name from the input. The answer is the value associated with feature *X*.
3) **Of the top *X* features, which are positive?** *X* is a random number between 2-5 inclusive. The task is to return the subset of the *X* most influential features that have a feature importance value greater than 0.
4) **Of the top *X* features, which are negative?** This follows the same pattern as above, but for feature importance values less than 0.
5) **Of these features [*ft_list*], which support the prediction?** *ft_list* is a list of 2-5 features, chosen at random from the input. The task is to return the subset of features from *ft_list* with a feature importance value greater than 0.
6) **Of these features [*ft_list*], which are against the prediction?** This follows the same pattern as above, but for feature importance values less than 0.
7) **Which features have an absolute value greater than *X*?** *X* is a random float between 0.30-0.45 inclusive. The goal is to return a list of features with a value above *X*.
8) **Which are the *X* least important features?** *X* is a random number between 2-5 inclusive. The task is to return a list of the *X* features with the lowest feature importance scores.

For questions 1 and 2, the answer is a single value, while the answers to questions 3-8 are lists of features separated by commas or blank if there is no correct answer.

VI. Results

A. Textual Explanation Generation

We fine-tune T5-base and BART-base models on the TEXEN and TEXEN-Augmented datasets. All experiments are run until validation performance has not increased for three epochs in a row. Once this limit has been reached, the best model is chosen, as decided by the lowest loss on the

TABLE III
ERROR ANALYSIS OF BART GENERATED TEXTUAL EXPLANATIONS PER SENTENCE TYPE. LOWEST ERROR RATE IN BOLD.

Experiment	Classification		Top		Unnamed Groups		Named Groups		Summary		Total	
	Count	Error Rate	Count	Error Rate	Count	Error Rate	Count	Error Rate	Count	Error Rate	Count	Error Rate
base-20	34	12%	29	14%	39	79%	17	53%	61	44%	180	42%
base-20-Aug	34	15%	30	**3%**	48	67%	39	41%	52	46%	203	38%
base-10	32	13%	30	13%	37	**49%**	25	52%	58	19%	182	27%
base-10-Aug	35	**6%**	30	10%	41	61%	29	62%	63	35%	198	35%
large-20	37	8%	32	22%	23	91%	42	43%	71	27%	205	33%
large-20-Aug	31	10%	29	7%	38	68%	21	**33%**	62	24%	181	29%
large-10	30	13%	34	15%	24	67%	31	42%	57	18%	176	27%
large-10-Aug	34	9%	29	14%	36	53%	30	40%	51	**14%**	180	**25%**

validation set. During inference, the neural generators generate textual explanations via beam search with values for the beam size, length penalty, and repetition penalty equal to 20, 1.6, and 1.5, respectively. Examples of generated narrations are shown in Fig. 6 and Fig. 7.

1) Automatic Evaluation: The quality of the output textual explanations is assessed using automatic metrics METEOR [40], BLEU [41], and BLEURT [15]. The BLEU and METEOR scores are employed to measure the surface-level similarity of the reference texts and the machine-generated text. On the other hand, the BLEURT score is a semantic equivalence-based metric that indicates how well the machine-produced text communicates the meaning of the reference text [15]. As a baseline for comparison, we translate the input into a fixed template style, similar to the model input but with values removed and set top_n as $min(n, 3)$:

> Predicted class is $<c_1>$, value of $<p_1>$. Other classes and values are $<c_2>$ $<p_2>$ & ... & $<c_m>$ $<p_m>$. Top features are [$<f_1>$, ..., and $<f_{top\_n}>$]. Positive features are [$<f_i^+>$, ..., and $<f_{top\_n^+}^+>$]. Negative features are [$<f_j^->$, ..., and $f_{top\_n^-}^->$]. Lowest impact features are [$<f_{n-4}>$, ..., and $<f_n>$].

The evaluation scores achieved by the models are shown in Table IV. We report the BLEU, BLEURT, and METEOR scores achieved on the test set. Compared to the baseline, all models show a significantly improved performance in all three reported metrics.

2) Error Analysis: We also conduct an error analysis on 30 records from the test set, generating narratives for each of our experiments and counting errors. Due to time constraints, we choose to focus on BART. Referring to Table V, (*base* / *large*) refers to BART-base or BART-large, (*10* / *20*) refers to top_n and *Aug* refers to the use of TEXEN-Augmented, as opposed to TEXEN. We sifted through each sentence of each narration, classifying them as either:

- **Classification**: Talking about the predicted class probability
- **Top features**: Mentioning the most influential features
- **Named groups**: Referring to positive, negative, moderately influential or least influential features
- **Unnamed groups**: Typically of the form "among these..." or "all the remaining features..."
- **Summary**: General statements summarizing the decision

If the sentence contained an error or did not make sense, then a one was tallied for that sentence, else zero. Table V, for each of the sentence types, shows, for each model, how many times each sentence appeared and the proportion of sentences of that type that contained an error across the 30 analyzed narrations. Analyzing the results, the model is more consistent at producing error-free sentences of certain types than others. "Classification", and "Top features" sentences are usually in a more consistent style in the collected narratives, which could be why the models were more successful at generating them. Using top_n of 10, rather than 20, tended to decrease the error rate, particularly in "Unnamed groups", which the models found difficult. As shown in Fig. 6 and Fig. 7, the Textual Explanation Generators struggled with specific phrases that grouped or excluded previously mentioned features and made a claim about the said group.

For all models except *base-10*, training on TEXEN-Augmented caused a lower error rate, demonstrating that providing more training data with re-randomized placeholders

TABLE IV
EVALUATION OF TEXTUAL EXPLANATION GENERATION PERFORMANCE OF THE NEURAL MODELS. AVG. RANK REFERS TO THE MEAN IN-COLUMN RANK. BEST IN-COLUMN SCORES ARE IN BOLD.

Experiment		BLEU	BLEURT	METEOR	Avg. Rank
BART	base-20	0.16	-0.25	0.36	6.7
	base-20-Aug	0.17	-0.23	0.36	**5.0**
	base-10	0.15	-0.19	0.34	8.0
	base-10-Aug	0.16	-0.23	0.36	6.3
	large-20	0.14	-0.28	**0.37**	9.3
	large-20-Aug	0.15	-0.27	0.35	10.7
	large-10	0.14	-0.25	0.34	12.7
	large-10-Aug	0.14	-0.26	0.35	10.3
T5	base-20	0.16	-0.22	0.34	8.7
	base-20-Aug	0.17	-0.27	0.35	6.3
	base-10	0.17	-0.31	0.35	9.0
	base-10-Aug	0.17	-0.28	0.35	8.3
	large-20	0.17	**-0.17**	0.34	7.7
	large-20-Aug	**0.18**	-0.37	0.34	9.7
	large-10	**0.18**	-0.22	0.34	5.3
	large-10-Aug	0.17	-0.34	0.34	11.3
Baseline		0.05	-0.80	0.19	17.0

TABLE V
QUESTION ANSWER RESULTS

Question	Accuracy	
	BART-base	**T5-base**
Value of class X?	94%	100%
Value of feature X?	87%	99%
Of top X, which are positive?	59%	85%
Of top X, which are negative?	76%	97%
Of ft_list, which support?	62%	90%
Of ft_list, which are against?	82%	92%
Which features are >X?	73%	87%
X least important features?	39%	73%
Total	73%	91%

allows the model to learn the input-narrative relationship more effectively and make fewer false claims. Using BART-large also yielded a lower error rate, most notably in "Summary" sentences where the generated narrative will tend to make broader statements without mentioning specific features, instead describing general patterns.

B. Question Answering

We train BART-base and T5-base models on the Question-Answer dataset and report the per-question accuracy in Table V. Analyzing the results, we can see that the models found some questions more straightforward: questions 1 and 2, which asked for a single class and feature value, scored the highest, perhaps because only a single figure was required instead of a list. For questions that need a list of numbers as an answer, if the generation matched the string exactly, then it was given a one, else zero. T5 scored especially highly, with an average accuracy of 91%. Examples are shown in Fig. 5

Q: Of the top 4 features, which are positive?
A: F5, F8, F1
T5 Pred: F5, F8, F1
BART Pred: F5, F8, C1
Q: Of the top 5 features, which are negative?
A: F4, F11
T5 Pred: F4, F11
BART Pred: F4, F11
Q: Which features have an absolute value greater than 0.38?
A:
T5 Pred:
BART Pred:
Q: Which are the 2 least influential features?
A: F8, F11
T5 Pred: F8, F11
BART Pred: F8, F11
Q: What is the value of F5?
A: 0.01
T5 Pred: 0.01
BART Pred: 0.01
Q: Of the top 5 features, which are positive?
A: F16, F19, F12, F17
T5 Pred: F16, F19, E12, C17
BART Pred: F16, F19, C12, f17
Q: Of these features [F1, F8, F10, F3], which support the prediction?
A: F1, F10, F3
T5 Pred: F1, F10, F3
BART Pred: F1, F10, C3
Q: Of the top 4 features, which are positive?
A: F8, F1, F7
T5 Pred: F8, F1, F7
BART Pred: F8, F1, C7
Q: What is the value of F3?
A: -0.05
T5 Pred: -0.05
BART Pred: -0.05
Q: What is the value of F17?
A: 0.01
T5 Pred: 0.01
BART Pred: 0.02
Q: Of the top 2 features, which are negative?
A: F7
T5 Pred: F7
BART Pred: [blank]
Q: Of these features [F4, F20, F30], which are against the prediction?
A: F20
T5 Pred: F20
BART Pred: F20

Fig. 5. Example of questions, reference answers and predictions from both models. Errors are in red.

VII. DISCUSSION

As demonstrated by these two tasks, our models are able to provide extra clarity to assist in machine learning interpretability. While further comparisons could strengthen our conclusions, our principal aim is to introduce the task and methodology. We recognize that some may consider the dataset small; however, the difficulty of collecting quality narrations meant it was very costly and time-consuming to generate. As a result, this dataset represents the largest possible dataset we had the means to collect, and we are pleased to make it publicly available to benefit other researchers in the field.

Our question-answer task was designed to cover the information held in numerical explanations; however, we acknowledge that the current set of questions may not cover all possible scenarios. Nevertheless, by using synthetic explanations, our dataset generation process allows for easy adaptation to encompass a new or expanded set of questions to suit specific needs.

VIII. CONCLUSION

In this work, we introduced a new NLG dataset of numerical-textual explanation pairs and trained T5 and BART to describe the output of feature importance-based explainers. When paired with the explainability graph, we aim to give users a better understanding of what the explanation means and, therefore, a better understanding of a given prediction decision. Automatic evaluation metrics show evidence of fluent explanations and error analysis yield reduced error rates when using TEXEN-Augmented. We also trained question-answer models for more structured answers and find T5-base gives us an overall accuracy of 91%. In the future, we plan to explore and utilize multi-modal modelling strategies, such as image captioning approaches, to directly use the explanation graphs without the linearization steps.

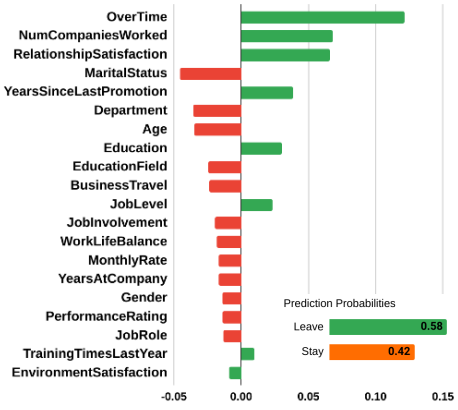
REFERENCES

[1] C. Longoni, A. Bonezzi, and C. K. Morewedge, "Resistance to Medical Artificial Intelligence," *Journal of Consumer Research*, vol. 46, no. 4, pp. 629–650, 2019. [Online]. Available: https://academic.oup.com/jcr/article/46/4/629/5485292

[2] J. K. Hentzen, A. Hoffmann, R. Dolan, and E. Pala, "Artificial intelligence in customer-facing financial services: a systematic literature review and agenda for future research," *International Journal of Bank Marketing*, vol. 40, no. 6, pp. 1299–1336, 2022. [Online]. Available: https://www.emerald.com/insight/0265-2323.htm

[3] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Information Fusion*, vol. 58, no. October 2019, pp. 82–115, 2020.

[4] European Commission, "Proposal for a Regulation of the European Parliament and of the Council Laying Down Harmonised Rules on Artificial Intelligence (Artificial Intelligence Act) and Amending Certain Union Legislative Acts," *Com(2021)*, vol. 0106, pp. 1–108, 2021.

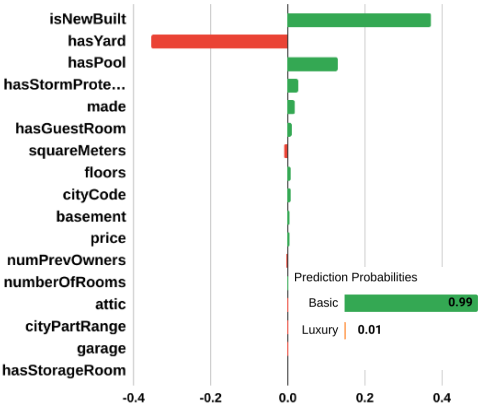


large-10-aug: The classifier is **57.83%** certain that **"Leave"** is the correct label for the case under consideration, implying that there is a **42.17%** chance that it could be **"Stay"** instead. The classification decision above is mainly based on the influence of features such as **OverTime**, **NumCompaniesWorked**, and **RelationshipSatisfaction**. Among these top influential features, only **MaritalStatus** has a negative impact, swinging the verdict in favour of any of the other labels. Other positive features with moderate influence include **Education**, **EducationField**, or **BusinessTravel**. However, considering the direction of effect of each input feature, it can be concluded that they are irrelevant to the prediction made here since their respective attributions are very close to zero. In simple terms, we can conclude that the model is not 100.0% confident in the assigned label given its level of certainty.

large-10: For the case under consideration, the model predicts class **"Leave"** with a probability of **57.83%**, implying that there is only a **42.17%** chance that **"Stay"** could be the correct label. The above prediction decision is mainly based on the values of **OverTime**, **NumCompaniesWorked**, and **RelationshipSatisfaction** which are shown to have very strong positive attributions, increasing the odds in favour of labelling the given case as **"Leave"**. On the contrary, all the remaining features such as **MaritalStatus**, **Department**, or **Age** negatively contribute to the decision made here. In terms of the direction of influence of each input feature, it can be concluded that the joint positive attribution outweighs the contributions of negative features hence the confidence level associated with the predicted label for this case.

Reference: 'The test case or example under consideration is labelled as **"Leave"** with a modest level of confidence because the probability that **"Stay"** is the correct label is approximately **42.17%**. The input features with the highest influence on the prediction above are **OverTime**, **NumCompaniesWorked**, **RelationshipSatisfaction**, **MaritalStatus**, **YearsSinceLastPromotion**, and **Department**, and those with moderate contributions are **Age**, **Education**, **EducationField**, **BusinessTravel**, **JobLevel**, **JobInvolvement**, and **WorkLifeBalance**. The influence of **MonthlyRate**, **YearsAtCompany**, **Gender**, **PerformanceRating**, **JobRole**, **TrainingTimesLastYear**, and **EnvironmentSatisfaction** can be described as moderately low. However, not all the features are considered by the classifier to arrive at the decision made for the given case. Irrelevant features include **YearsWithCurrManager**, **DailyRate**, **YearsInCurrentRole**, and **TotalWorkingYears**. The positive features driving the prediction in favour of the predicted label are **OverTime**, **NumCompaniesWorked**, **RelationshipSatisfaction**, **YearsSinceLastPromotion**, **Education**, **JobLevel**, and **EnvironmentSatisfaction**. Overall, the majority of the influential features have negative attributions that decrease the probability that **"Leave"** is the correct label, explaining the uncertainty associated with the prediction decision made by the classifier.'

Fig. 6. Example of generated textual explanations and reference narration. The numerical explanation is shown here graphically on the left. The numerical explanation has feature and class names randomized and is fit into a template, as described in Section IV-C1. Top right and center show the output of two trained textual explainers, the difference being that large-10-aug has been trained on TEXEN-Augmented and large-10 trained on TEXEN. At the bottom, the reference text is shown. This is the narrative that was collected when the annotator was provided with the graph in this figure. Features, classes and values are in bold. Errors are in red.



large-10-aug: The classifier is **99.30%** certain that the correct label for this case is **"Basic"**, implying that there is only a **0.70%** chance that **"Luxury"** could be the true label. The classification decision above is mainly based on the influence of features such as **isNewBuilt**, **hasYard**, and **hasPool**. On the other hand, some of the least relevant features are shown to have negative attributions, shifting the verdict in the opposite direction. These negative features include **floors**, **cityCode**,and **basement**. However, considering the predicted likelihoods across the classes, it can be concluded that all the remaining features positively contribute to the prediction made here. As a matter of fact, the top positive features increasing the odds in favour of labelling the given case as **"Basic"**. In contrast, those with moderate contributions contradicting the model's decision regarding the appropriate label are **price**, or **cityPartRange**. To put it concisely, when you take into account the very high degree of certainty associated with the assigned label, one has to look at the attribution analysis to understand why the confidence level is quite high.

large-10: For the case under consideration, the model predicted **"Basic"** with **99.30%** certainty, implying that there is only a **0.70%** chance that **"Luxury"** could be the correct label. The prediction decision above is mainly based on the influence of the following features: **isNewBuilt**, **hasYard**, and **hasPool** which are shown to have very strong positive attributions, increasing the odds of labelling the given case as **"Basic"**. On the contrary, all the remaining features such as **hasStormProtector**, **made**, or **hasGuestRoom** negatively contribute to the prediction made here. All in all, it is valid to conclude that the classifier is very certain about the assigned label considering the degree of impact of each input feature.

Reference: Considering the values of the input features, the classifier generates the label **"Basic"** with **close to 100%** confidence, since the prediction probability of **"Luxury"** is only **0.70%**. The above classification judgement is mainly due to the influence of the features **isNewBuilt**, **hasPool**, and **hasYard** mainly because the classifier places more emphasis on their values than the remaining ones. Among these top features, **hasYard** is the one exhibiting negative influence, shifting the prediction decision towards the least probable class, **"Luxury"** and away from **"Basic"**. Conversely, **isNewBuilt** and **hasPool** are referred to as positive features since they increase the odds of the assigned **"Basic"** label instead of **"Luxury"**. Finally, unlike all the aforementioned, the values of **attic**, **cityPartRange**, **garage**, and **hasStorageRoom** have little impact on the classification output decision made here.

Fig. 7. Another example of generated textual explanations and reference narration. Same format as in Fig. 6.

[5] C. Molnar, *Interpretable Machine Learning. A Guide for Making Black Box Models Explainable.*, 2019. [Online]. Available: https://christophm.github.io/interpretable-ml-book

[6] D. Castelvecchi, "Can we open the black box of AI?" *Nature*, vol. 538, no. 7623, pp. 20–23, 2016.

[7] C. Rudin and J. Radin, "Why Are We Using Black Box Models in AI When We Don't Need To? A Lesson From an Explainable AI Competition," *Harvard Data Science Review*, vol. 1, no. 2, 2019. [Online]. Available: https://hdsr.mitpress.mit.edu/pub/f9kuryi8

[8] M. T. Ribeiro, S. Singh, and C. Guestrin, "" Why should i trust you?" Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.

[9] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 4765–4774.

[10] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *International conference on machine learning*. PMLR, 2017, pp. 3319–3328.

[11] A. Binder, S. Bach, G. Montavon, K.-R. Müller, and W. Samek, "Layer-wise relevance propagation for deep neural network architectures," in *Information science and applications (ICISA) 2016*. Springer, 2016, pp. 913–922.

[12] B. Peng, C. Zhu, C. Li, X. Li, J. Li, M. Zeng, and J. Gao, "Few-shot Natural Language Generation for Task-Oriented Dialog," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, 2020, pp. 172–182.

[13] Y. Su, Z. Meng, S. Baker, and N. Collier, "Few-Shot Table-to-Text Generation with Prototype Memory," in *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2021, pp. 910–917.

[14] L. H. Suadaa, H. Kamigaito, K. Funakoshi, M. Okumura, and H. Takamura, "Towards table-to-text generation with numerical reasoning," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021, pp. 1451–1465.

[15] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," *Journal of Machine Learning Research*, vol. 21, pp. 1–67, 2020.

[16] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 7871–7880.

[17] M. Strauss and M. Kipp, "Eric: a generic rule-based framework for an affective embodied commentary agent," in *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 1*, 2008, pp. 97–104.

[18] T.-H. Wen, M. Gasic, N. Mrksic, P.-h. Su, D. Vandyke, and S. J. Young, "Semantically Conditioned LSTM-based Natural Language Generation for Spoken Dialogue Systems," in *EMNLP*, 2015.

[19] T. Liu, K. Wang, L. Sha, B. Chang, and Z. Sui, "Table-to-text generation by structure-aware seq2seq learning," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[20] A. Gatt and E. Krahmer, "Survey of the state of the art in natural language generation: Core tasks, applications and evaluation," *Journal of Artificial Intelligence Research*, vol. 61, pp. 65–170, 2018.

[21] R. Perera and P. Nand, "Recent advances in natural language generation: A survey and classification of the empirical literature," *Computing and Informatics*, vol. 36, no. 1, pp. 1–32, 2017.

[22] Y. Yang, J. Cao, Y. Wen, and P. Zhang, "Table to text generation with accurate content copying," *Scientific reports*, vol. 11, no. 1, pp. 1–12, 2021.

[23] E. Goldberg, N. Driedger, and R. I. Kittredge, "Using natural-language processing to produce weather forecasts," *IEEE Expert*, vol. 9, no. 2, pp. 45–53, 1994.

[24] C. van der Lee, E. Krahmer, and S. Wubben, "PASS: A Dutch data-to-text system for soccer, targeted towards specific audiences," in *Proceedings of the 10th International Conference on Natural Language Generation*, 2017, pp. 95–104.

[25] E. Reiter and R. Dale, "Building applied natural language generation systems," *Natural Language Engineering*, vol. 3, no. 1, pp. 57–87, 1997.

[26] H. Harkous, I. Groves, and A. Saffari, "Have Your Text and Use It Too! End-to-End Neural Data-to-Text Generation with Semantic Fidelity," in *Proceedings of the 28th International Conference on Computational Linguistics*, 2020, pp. 2410–2424.

[27] A. Parikh, X. Wang, S. Gehrmann, M. Faruqui, B. Dhingra, D. Yang, and D. Das, "ToTTo: A Controlled Table-To-Text Generation Dataset," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 1173–1186.

[28] H. Wang, X. Zhang, S. Ma, X. Sun, H. Wang, and M. Wang, "A neural question answering model based on semi-structured tables," in *Proceedings of the 27th International Conference on Computational Linguistics*, 2018, pp. 1941–1951.

[29] W. Chen, M.-W. Chang, E. Schlinger, W. Y. Wang, and W. W. Cohen, "Open Question Answering over Tables and Text," in *International Conference on Learning Representations*, 2020.

[30] S. Chemmengath, V. Kumar, S. Bharadwaj, J. Sen, M. Canim, S. Chakrabarti, A. Gliozzo, and K. Sankaranarayanan, "Topic Transferable Table Question Answering," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021, pp. 4159–4172.

[31] L. F. R. Ribeiro, M. Schmitt, H. Schütze, and I. Gurevych, "Investigating Pretrained Language Models for Graph-to-Text Generation," in *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*. Online: Association for Computational Linguistics, 11 2021, pp. 211–227. [Online]. Available: https://aclanthology.org/2021.nlp4convai-1.20

[32] R. Koncel-Kedziorski, D. Bekal, Y. Luan, M. Lapata, and H. Hajishirzi, "Text Generation from Knowledge Graphs with Graph Transformers," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 2284–2293.

[33] W. Chen, J. Chen, Y. Su, Z. Chen, and W. Y. Wang, "Logical Natural Language Generation from Open-Domain Tables," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 7929–7942.

[34] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, and others, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[35] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "{BERT}: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of the 2019 Conference of the North {A}merican Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, 6 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423

[36] J. M. Alonso, P. Ducange, R. Pecori, and R. Vilas, "Building explanations for fuzzy decision trees with the expliclas software," in *2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 2020, pp. 1–8.

[37] J. M. Alonso and A. Bugar\'\in, "ExpliClas: automatic generation of explanations in natural language for weka classifiers," in *2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 2019, pp. 1–6.

[38] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[39] A. Moryossef, Y. Goldberg, and I. Dagan, "Step-by-Step: Separating Planning from Realization in Neural Data-to-Text Generation," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 2267–2277.

[40] S. Banerjee and A. Lavie, "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments," in *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, 2005, pp. 65–72.

[41] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting on association for computational linguistics*. ACL, 2002, pp. 311–318.