IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 8 April 2025, accepted 1 May 2025, date of publication 6 May 2025, date of current version 21 May 2025.
Digital Object Identifier 10.1109/ACCESS.2025.3567566

RESEARCH ARTICLE

Enhancing ECG Report Generation With Domain-Specific Tokenization for Improved Medical NLP Accuracy

FARZEEN ASHFAQ[1], NZ JHANJHI[1], (Member, IEEE), NAVID ALI KHAN[1], DANISH JAVED[1], MEHEDI MASUD[2], (Senior Member, IEEE), AND MOHAMMAD SHORFUZZAMAN[2], (Senior Member, IEEE)
[1]School of Computer Science, Taylor's University, Subang Jaya 47500, Malaysia
[2]Department of Computer Science, College of Computers and Information Technology, Taif University, Taif 21944, Saudi Arabia
Corresponding author: NZ Jhanjhi (noorzaman.jhajhi@taylors.edu.my)

This research was funded by Taif University, Saudi Arabia, project number (TU-DSPP-2024-04).

ABSTRACT The automation of medical report generation has been an area of interest for researchers over the years, with significant advancements in computational techniques and natural language processing. Traditionally, the focus has remained on rule-based or template-driven approaches to streamline the documentation process. However, recent breakthroughs in generative AI models have substantially enhanced the capabilities of automated medical report generation. Hence, the field has gained significant attention in recent years due to its potential to support healthcare professionals and improve diagnostic workflows. One key challenge in these models is the reliance on pre-trained, general-purpose tokenizers, which often fail to capture the domain-specific vocabulary essential for accurate medical reporting. When using a general tokenizer, the generated reports may lack coherence, relevance, or even correct medical terminology, leading to poor quality outputs. This problem is particularly acute in specialized fields like ECG, where precise terminology is critical. To address this gap, we train a Byte Pair Encoding (BPE) tokenizer to incorporate ECG-specific vocabulary, resulting in improved coherence and relevance in text generation. The custom tokenizer was integrated into a GPT-2 model for image-to-text generation tasks, where ECG reports were generated from ECG waveforms. Our experiments show that using the custom tokenizer leads to a 85% improvement in coherence and a 45% increase in relevance of generated reports compared to a general-purpose tokenizer.

INDEX TERMS Fine-tuning, GPT-2, image-to-text conversion, ECG report generation, custom tokenizer.

I. INTRODUCTION

Medical report generation has become a critical task in healthcare [1], with the increasing need to streamline documentation and support healthcare professionals in diagnostic workflows. Traditional approaches, such as rule-based or template-driven methods [2] and [3], have limitations in capturing the complexity and variability of medical language. With the rise of generative AI models, the potential for automated medical report generation has greatly improved, offering more dynamic and context-sensitive solutions. In recent years natural language processing has made significant success in every domain with models like BERT [4], GPT-2 [5] and other large pre-trained transformers. These models perform exceptionally well on wide range of general language tasks. highlights the shift from a static, template-based approach to a more dynamic, AI-driven approach in report generation [6]. Figure 1 highlights the shift from a static, template-based approach to a more dynamic, AI-driven approach in report generation. The AI-driven method allows for more flexible, context-sensitive, and potentially more insightful reports by utilizing data analysis and machine

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

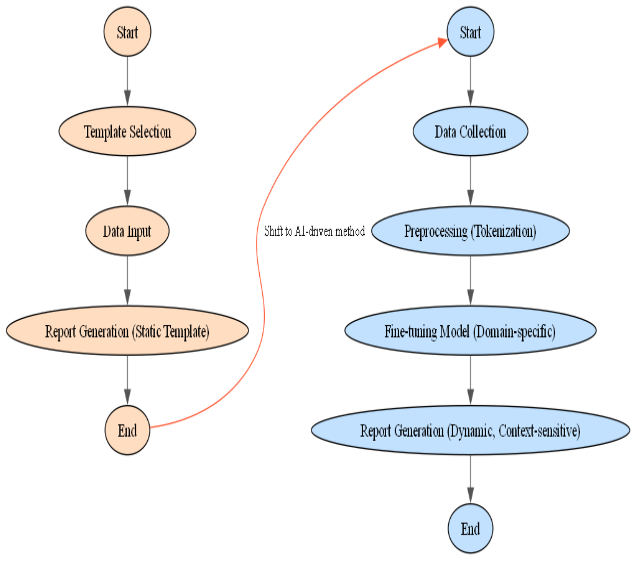


FIGURE 1. Workflow comparison between two methods of report generation.

learning. The AI-driven method allows for more flexible, context-sensitive, and potentially more insightful reports by using. data analysis and machine learning. However, when it comes to specialized domains such as medical language, biomedical reports such as ECG report generation, the usage of these models without domain specific fine-tuning or adaptation often leads to various challenges [28].

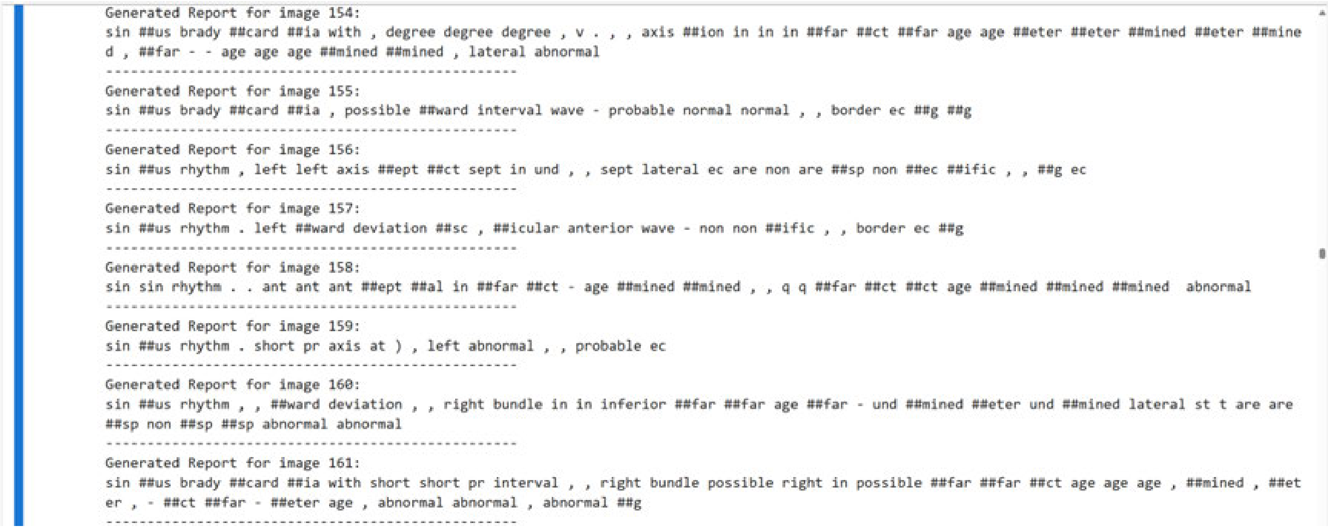
A. CHALLENGES IN USING GENERAL PURPOSE TOKENIZERS FOR MEDICAL REPORT GENERATION

Tokenizers are essential components in natural language processing systems [7]. They break down text into smaller units called tokens, which could be words, subwords, or characters. These tokenizers are used for various tasks such as language modeling [8], machine translation, sentiment analysis, and more. But when it comes to some specific domain particularly medical domain, medical text generation using a general-purpose tokenizers can introduce numerous challenges due to the specialized nature of medical language.

1) **Vocabulary Mismatch:** BERT, GPT-2, and Similar Models are trained on large corpora of general text from a variety of domains (e.g., books, articles, websites, etc.). These models learn a vocabulary that is highly suited to handling general language but lacks specialized terms relevant to the medical field. Medical terms, including those specific to ECG (electrocardiogram) readings, such as hypertrophy, repolarization, supraventricular etc and others, are often underrepresented or not present at all in the pre-trained vocabulary. Hence, when a model like GPT-2 or BERT is applied to the ECG report generation task, it may fail to properly recognize or understand these terms. This results in the generation of irrelevant, incorrect, or fragmented text that doesn't align with the structure or content required for a professional medical report.
2) **Contextual Understanding of Medical Data:** Medical Text has a very specific structure. ECG reports typically follow a formalized template, describing the patient's condition, the results of the ECG, and often suggestions for further investigation or treatment. The context within which terms are used matters greatly. For instance, the QT interval in an ECG report is a specific measure that requires precise context for it to be interpreted correctly.
3) **Suboptimal Tokenization of Rare or Domain-Specific Terms:** General-purpose tokenizers work by breaking words into subword units (like Byte Pair Encoding (BPE) or WordPiece). While this helps handle out-of-vocabulary words, the subwords produced may not be meaningful in the context of specialized medical language. Tokenizers like BERT's WordPiece or GPT-2's BPE might break medical terms into fragments. For instance, "tachycardia" might be broken down into ["tach", "##y", "##card", "##ia"], which is not ideal for generating fluent, medically accurate text. The model would then struggle to form coherent medical sentences and may generate incoherent or incorrect reports due to these fragmented tokens. The example of using non customized tokenizer is shown in figure 2 with the broken words being observed.

Hence, we find the as a research problem that in the context of ECG image-to-text generation, using a non-custom tokenizer presents a multitude of challenges. From vocabulary mismatches to poor handling of medical terminology and report structure, general-purpose tokenizers like those used in BERT or GPT-2 are not designed to process the unique requirements of medical language. They can lead to fragmented text, misinterpretations of key terms, and reports that lack medical accuracy, coherence, and structure. Therefore, we propose the adaptation of a custom tokenizer trained specifically on ECG reports, enabling more accurate and coherent text generation. By fine-tuning the tokenizer to understand ECG-specific language, our approach ensures that the model generates medically relevant reports. Our results show significant improvements in text relevance and coherence, demonstrating faster model convergence in image-to-text generation tasks. This work highlights the importance of domain-specific tokenization in improving automated medical report generation, contributing to the development of more efficient and reliable AI systems in healthcare. This research work will open a new direction for the ECG reports by contributing at this stage as below:

1) Our research contributions demonstrate that domain-specific tokenization significantly enhances the quality of text generation in medical contexts, particularly for ECG reports. By prioritizing ECG-specific terminology, we improved the model's accuracy and its ability



```
Generated Report for image 154:
sin ##us brady ##card ##ia with , degree degree degree , v . , , axis ##ion in in in ##far ##ct ##far age age ##eter ##eter ##mined ##eter ##mine
d , ##far - - age age age ##mined ##mined , lateral abnormal
--------------------------------------------------
Generated Report for image 155:
sin ##us brady ##card ##ia , possible ##ward interval wave - probable normal normal , , border ec ##g ##g
--------------------------------------------------
Generated Report for image 156:
sin ##us rhythm , left left axis ##ept ##ct sept in und , , sept lateral ec are non are ##sp non ##ec ##ific , , ##g ec
--------------------------------------------------
Generated Report for image 157:
sin ##us rhythm . left ##ward deviation ##sc , ##icular anterior wave - non non ##ific , , border ec ##g
--------------------------------------------------
Generated Report for image 158:
sin sin rhythm . . ant ant ant ##ept ##al in ##far ##ct - age ##mined ##mined , , q q ##far ##ct ##ct age ##mined ##mined ##mined  abnormal
--------------------------------------------------
Generated Report for image 159:
sin ##us rhythm . short pr axis at ) , left abnormal , , probable ec
--------------------------------------------------
Generated Report for image 160:
sin ##us rhythm , , ##ward deviation , , right bundle in in inferior ##far ##far age ##far - und ##mined ##eter und ##mined lateral st t are are
##sp non ##sp ##sp abnormal abnormal
--------------------------------------------------
Generated Report for image 161:
sin ##us brady ##card ##ia with short short pr interval , , right bundle possible right in possible ##far ##far ##ct age age age , ##mined , ##et
er , - ##ct ##far - ##eter age , abnormal abnormal , abnormal ##g
--------------------------------------------------
```

FIGURE 2. **The image-to-text report model's output struggling to form coherent medical sentences while generating incoherent or incorrect reports due to the fragmented tokens.**

to generate contextually relevant and medically precise text.

2) In our ECG image-to-text research, implementing a custom tokenizer ensured that the generated text aligned seamlessly with established medical vocabulary, thereby enhancing the practical applicability and reliability of such systems in real-world clinical settings.
3) Furthermore, our work underscores the importance of training language models for specialized applications, such as medical report generation. We demonstrated how targeted tokenization strategies can effectively adapt a model, specifically GPT-2 to a specific domain's unique vocabulary and requirements, paving the way for more accurate and domain-aware AI systems in healthcare.

We would also like to mention here that this work is part of a broader research focus aimed at advancing ECG image-to-text generation, where the goal is to train a GenAI model to take as an input direct ECG waveform (12-Lead) image and transform the image into coherent and medically relevant diagnostic report. In this context, the development of a custom tokenizer adapted specifically for ECG terminology plays a pivotal role in enhancing both the quality of the generated text and the efficiency of the model training process. In our other part of the research once we experimented with both pretrained and our custom tokenizer (ECG-specific language), we observed that the later enabled the model to produce more accurate and contextually appropriate reports. Moreover, the custom tokenizer accelerated the convergence of the model during training, allowing for quicker and more effective learning. This adaptation not only improved the overall performance of the image-to-text generation task, but we feel that it can also contribute to wider domains and research problems such as radiology, pathology, and other specialized medical fields, leading to more reliable and medically sound automated reporting systems in healthcare.

The remaining paper is organized as follows: Section II presents a detailed review of related works, highlighting key studies and developments in the field. Section III outlines the methodology employed in this study, detailing the research design, techniques, and processes used to achieve the results. Section IV provides a comprehensive analysis of the results, including key findings and data interpretations. In Section V, the discussion explores the implications of the results, comparing them with existing literature, and addressing the significance of the findings. Finally, Section VI concludes the paper, summarizing the main points and suggesting potential areas for future research.

II. RELATED WORKS

One of the research areas that is receiving increasing attention is the automated medical report generation task derived from various types of data, such as medical images, sensor signals, or patient records. This research holds promise in improving healthcare workflows by reducing the time required for documentation and improving clinical decision-making. The early approaches to automated reporting were largely based on rule-based systems relying on predefined templates [9], [10]. These methods, while useful, were limited in their ability to handle the complexities and variability of medical language. Advances in machine learning models have greatly improved the ability of models to generate human like text. Then recently in introduction of neural network based architectures including RNN, LSTM and transformers have become more powerful models for every domain particularly for medical text generation and image captioning task.

The introduction of image-to-text generation models, particularly in medical fields, has brought about significant advancements. One area of notable research is in radiology, where automated systems are being developed to generate diagnostic reports from medical images such as X-rays, CT scans, and MRIs. CDGPT proposed by [11], uses pre-trained ChexNet model to predict abnormalities from chest x-ray images and then conditioned a pre-trained GPT2 model on both the visual features (from the x-ray image) and the semantic features (from the tags) to generate the full radiology report. Similarly, [12] uses a multimodal approach with a hybrid CNN-LSTM to generate diagnostic radiology reports. The authors in [13] and [14] uses a transformer based architecture for generating medical reports, specifically in the context of Chest X-ray images. Recently, [15] addresses the limitations of previous models by incorporating medical knowledge and improving the interaction between images and text in radiology report generation.

Despite a number of research being held in radiology image to text report generation ECG (electrocardiogram) image to report generation has received comparatively less attention. The possible reasons could be:

1) While radiology images (e.g., X-rays, CT scans, MRIs) provide visual data that can be interpreted in a more standardized manner [16], ECG signals are more abstract. The analysis of ECGs often involves both temporal and spectral aspects, requiring a high understanding of various heart conditions [17]. This can be more challenging to automate effectively, as the patterns may not be as visually consistent as those in medical images.
2) Radiology datasets, such as those used for X-rays or CT scans, are often more readily available and larger in size compared to ECG datasets. The structured nature of ECG signals does not necessarily lend itself to easily obtaining the large-scale annotated datasets [18] required for deep learning models to generalize effectively.
3) Radiology reports tend to follow more standardized formats, which makes them easier to automate. ECG reports, however, may vary in the level of detail or terminology [19], making it more difficult for AI to generate them in a consistent and clinically useful manner.
4) Transformers and similar architectures are highly effective in image-based tasks (such as radiology) because of their ability to learn complex spatial relationships [20], [21]. While there have been couple of excel in ECG classification using deep learning, the signal nature of ECG data (one-dimensional time series) differs from the two-dimensional nature of medical images, posing a unique challenge in terms of applying the same techniques used in image data to ECGs.

Reference [22] proposes ECG caption generation method employing an encoder-decoder architecture, using a ResNet-based encoder to embed ECG signals and either Transformer or LSTM decoders for generating descriptive reports. The Transformer decoder uses self-attention to dynamically focus on different ECG segments, while the LSTM decoder leverages attention mechanisms to weight ECG segments, generating clinically relevant descriptions of the ECG data. The authors in [23] adapted CLIP model to process 3D ECG data processed with wavelet scattering. The authors names their model cardioGPT where the model simply takes the input ECG signal convert to wavelet scattering network and then the GPT is given the input to generate textual report descriptions.

In addition to the works presented in [22] and [23], several other studies have contributed to the field of ECG image-to-text generation. These studies, including various methods for ECG signal embedding, report generation, and model architectures, are summarized in Table 1. This table provides a comparative overview of the approaches, highlighting key differences and innovations in each study.

While the aforementioned studies have made significant success in improving ECG image-to-text generation by exploring various embedding techniques, model architectures, and attention mechanisms, they still often rely on pre-trained, general-purpose tokenizers for processing the generated text. These tokenizers, which are typically trained on vast amounts of general-purpose data, are often not trained to understand the specialized vocabulary and domain-specific terms necessary for high-quality, medically relevant reports. This limitation becomes particularly evident when dealing with highly specialized areas like ECG, where precise terminology and nuanced medical concepts are critical.

In this context, general-purpose tokenizers may struggle with handling medical jargon, abbreviations, or specific medical symbols, resulting in less coherent or inaccurate report generation. Thus, there is a growing need for tokenization strategies tailored to these domains to improve the overall quality and relevance of the generated text. The importance of such domain-specific tokenization becomes increasingly evident as we move toward automated medical report generation, where even small inaccuracies can have significant consequences for patient care and clinical decision-making.

III. METHODOLOGY

In this section we outlines the methodology employed for developing and integrating a custom tokenizer for ECG report generation, incorporating GPT-2 for text generation, and evaluating the performance of the model. The entire pipeline from data preprocessing to the training of the tokenizer and model is detailed below. Figure 3 shows our pipeline for pretraining ECG-domain specific text generation model.

A. CUSTOM TOKENIZER DEVELOPMENT

The goal of the custom tokenizer is to efficiently tokenize ECG report texts, which are often domain-specific and contain medical terms. The process of developing the custom tokenizer can be broken down into the following steps:

1) DATA SELECTION

For this we used MIMIC IV physionet ecg machine generated text reports [29], [30].

TABLE 1. Comparative summary of studies on ECG image-to-text report generation.

Study	Methodology	Key Features	Model Architecture	Literature Findings
[22]	Encoder-decoder approach for ECG captioning	ResNet-based encoder for embedding ECG, Transformer or LSTM decoder for reports	ResNet + Transformer/LSTM	Uses attention mechanisms to dynamically focus on ECG segments
[23]	Adapted CLIP model for 3D ECG data	Wavelet scattering preprocessing, GPT for text generation	CLIP + GPT	CardioGPT generates textual reports from wavelet-transformed ECG signals
[24]	Vision encoder-decoder model for ECG image captioning	1.4 million ECGs and cardiologist diagnoses used to train the model for diverse cardiovascular conditions	BEiT transformer encoder + GPT-2 decoder	Generates ECG reports from images
[25]	Vision encoder-decoder model (ECG-GPT)	2.6 million ECGs across multiple health settings	Vision encoder-decoder with transformer	generating expert-level reports from ECG images
[27]	ECG Encoder-MLP Adapter	Aligns ECG features with text reports using a contrastive learning architecture.	Encoder-Decoder	A final LLM is constructed to generate detailed ECG diagnosis report

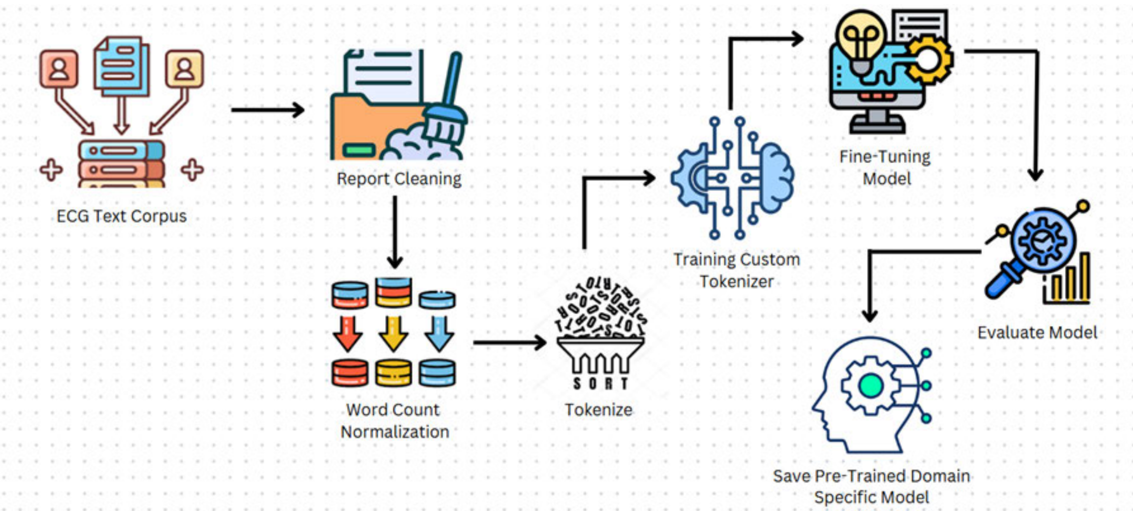


FIGURE 3. Pipeline for pretraining a domain specific model for generating ECG text data.

2) TEXT PREPROCESSING

We cleaned the ECG reports contain special characters (like "#NAME" @ and * etc.) and numbers that are irrelevant for tokenization. We remove non-alphanumeric characters (e.g., punctuation) using regular expressions. The text is also converted to lowercase to ensure consistency in tokenization. The tokenizer is based on the Byte Pair Encoding (BPE) model, which is well-suited for handling rare and out-of-vocabulary words. The BPE method works by iteratively merging frequent pairs of characters or subwords into new tokens, effectively creating a vocabulary of subword units that are more robust to unseen words in the report corpus. The primary cleaning steps included:

- Removing special characters: ECG reports often contain symbols and special characters that do not contribute meaningfully to the model. Using regular expressions, we filtered out punctuation and unwanted symbols.
- Eliminating numbers: Many ECG reports contain patient-specific numerical values (e.g., heart rate, interval durations), which are not relevant for text generation. Removing these ensures the model focuses on learning linguistic patterns rather than memorizing numerical values.
- Stopword removal: Some common words (e.g., "the", "is", "of") do not carry significant meaning in ECG reports and were removed to streamline the vocabulary.
- Lowercasing: To maintain consistency, all text was converted to lowercase, preventing redundant token variations (e.g., "ECG" vs. "ecg").

As a preprocessing step we also did word count normalization to make sure that the text reports are standardized to same length. Since Transformers require fixed input dimensions, this step ensured that all text reports had a standardized length before feeding them into the model. We accomplished this by:

- Truncating long reports to a fixed length (e.g., 40 words).

- Padding shorter reports with a <pad> token to maintain uniform input size.

This normalization was essential because GPT-2 uses attention mechanisms that depend on sequence length. Without this step, reports of varying lengths could lead to unstable training and inefficient memory usage.

3) TOKENIZATION AND VOCABULARY EXPANSION

In order to adapt the model to the domain-specific task of ECG report generation, we modified the tokenizer to include a set of special tokens. The tokenizer was trained using BPE on the ECG report dataset, and we appended the following special tokens to the vocabulary:

1) $< s >$: Start of the sentence token
2) $< pad >$: We used the padding token for padding sequences to a fixed length.
3) $< unk >$: The unknown token was used for handling out-of-vocabulary words.
4) $< mask >$: The mask token was used for masked language modeling tasks.

These tokens were initialized with random weights, as they were not present in the original tokenizer vocabulary. To ensure that the model's embeddings can accommodate these new tokens, we resized the model's embedding matrix using $model.resize\_token\_embeddings(len(tokenizer))$. During the training process, the model learned useful representations for these new tokens through backpropagation, updating their embeddings as part of the optimization process. By adding these domain specific tokens, the model could better handle padding, unknown tokens, and sequence boundaries, which are crucial for generating coherent ECG reports. During the fine-tuning of the pre-trained GPT-2 language model, the newly added tokens are incorporated into the model's vocabulary, and their embeddings, initially initialized with random weights, are updated through training. AAs the model performs the task of next-word prediction, the weights associated with the newly added tokens are learned and adjusted based on the data, allowing the model to adapt to the specific domain language (ECG reports). This approach enables the model to handle the custom vocabulary effectively, and through fine-tuning, it improves its understanding and generation of domain-specific text.

4) TRAINING THE TOKENIZER

The tokenizer is trained on a corpus of preprocessed ECG reports. This involves creating a vocabulary where each unique subword corresponds to a token. Special tokens are also defined to handle padding ([PAD]), start (<s>), end (</s>), unknown (<unk>), and masked (<mask>) tokens. These tokens are necessary for downstream tasks like text generation and sequence-to-sequence learning. The tokenization process follows the BPE algorithm, which starts by counting the frequency of all character pairs in the corpus. The most frequent pairs are merged into a single token, and this process is repeated until the desired vocabulary size is reached.

5) FINAL TOKENIZER CONFIGURATION

After training, the tokenizer is saved in a JSON format and is ready to be used with models like GPT-2 for text generation. The tokenizer also allows for padding and truncation of input sequences to a fixed length, ensuring compatibility with the model architecture.

B. INTEGRATION WITH GPT-2 FOR TEXT GENERATION

The custom tokenizer is integrated with GPT-2, a pre-trained causal language model. The integration process involves the following steps:

1) MODEL SELECTION

GPT-2 is selected due to its proficiency in autoregressive text generation, which works well for generating coherent sequences based on a given prompt. The model is pre-trained on large-scale textual data and can generate fluent and contextually relevant text given a sequence of tokens.

2) TOKENIZER MODEL ALIGNMENT

The vocabulary size of the custom tokenizer is different from the original GPT-2 tokenizer. To align the tokenizer with GPT-2's architecture, the model's token embeddings are resized according to the custom tokenizer's vocabulary size. This ensures that the model can correctly interpret the new tokens generated by the custom tokenizer.

3) FINE-TUNING

After creating the custom tokenizer, we load a pre-trained GPT-2 language model and fine-tune this model on the ECG reports using the newly created custom tokenizer. Fine-tuning the GPT-2 model with the ECG report dataset is necessary to adapt the model to the domain specific language of ECG reports. During fine-tuning, the model learns to generate coherent text based on ECG report inputs, maintaining contextual relevance to the medical terminology used in the reports.

Through this, the model learns to incorporate ECG-specific terms that were added during the tokenizer modification process. This ensured the model generates reports that include accurate and contextually appropriate medical language, reducing the risk of generating incorrect or ambiguous terminology.

The custom tokenizer plays a key role in making the fine-tuning process more efficient. By incorporating domain-specific tokens, the model can focus on learning ECG-specific patterns and terminology early on, resulting in faster convergence and improved performance on specialized tasks compared to models trained with generic tokenizers.

C. EVALUATION METHOD

To assess the effectiveness of the custom tokenizer and the text generation process, we evaluate the model on several key metrics:

1) TRAINING AND VALIDATION PROCEDURE

The training procedure involves feeding the tokenized ECG reports into the GPT-2 model, which learns to generate sequences of text based on the given input. The dataset is divided into training, validation, and test sets, with 80% of the data used for training, 10% for validation, and 10% for testing.

During training, the model is optimized using the Adam optimizer with a learning rate of 1e-5. The loss function used is cross-entropy loss, which measures the difference between the predicted token probabilities and the actual target tokens.

2) EVALUATION METRICS

- Perplexity: Perplexity is a measure of how well the model predicts the next token in a sequence. It is calculated as the exponentiation of the average negative log-likelihood of the predicted tokens:

$$Perplexity = e^{\frac{1}{N}\sum_{i=1}^{N} \log P(y_i|x_{i-1})} \tag{1}$$

  where $P(y_i|x_{i-1})$ is the probability of token, y_i given the previous tokens x_{i-1} and N is the number of tokens in the sequence.
- BLEU Score: The BLEU score is a metric for evaluating the quality of text generation by comparing the generated output to reference outputs. It is commonly used in machine translation tasks. The BLEU score is calculated using precision metrics for n-grams, where higher values indicate better quality text generation.
- The ROUGE score (Recall-Oriented Understudy for Gisting Evaluation) is used to evaluate the quality of summaries and text generation tasks by comparing the overlap of n-grams, words, and word sequences between the generated text and reference texts.

3) TRAINING LOOP

The training loop follows a standard procedure of iterating over the dataset, making forward passes through the model, computing the loss, and updating the model parameters via backpropagation. The training process also includes periodic validation to track model performance on unseen data.

4) METRICS COLLECTION

During training, we track the training and validation losses at each epoch, as well as BLEU and ROUGE scores. The training loss is computed as the cross-entropy loss between the predicted token sequences and the target sequences. The validation loss is similarly computed but on the validation set, providing an indication of how well the model generalizes.

A final evaluation is performed on the test set to obtain the final metrics. This is done after training completes and the model is fully fine-tuned.

D. BASELINE ARCHITECTURE

For the baseline architecture we selected one of the Hugging Face's library fast tokenizers that named GPT-2 to fine tune on ECG domain vocabulary. This serves as the text generation pipeline in our image-to-text component research as mentioned earlier. The architecture in the figure 4 depicts the most relevant parts of the transformer model for our purpose the attention mechanism and the stack of layers used to encode the input sequences is based solely on the decoder part of the original transformer. This decoder is responsible for generating output tokens. It generates text by predicting the next token, which can be a word or sub-word based on the context provided by previous tokens. GPT-2 uses a stack of transformers each block in it consists of two main sub-layers.

The algorithm for the custom tokenizer development, model integration, and training is shown in algorithm 1.

Algorithm 1 Custom Tokenizer and GPT-2 Model Integration for ECG Report Generation

Input: ECG reports dataset `ecg_reports.csv`
Output: Trained GPT-2 model and tokenizer
Load and preprocess ECG reports:
  `preprocess_text(text)` to remove special characters and convert text to lowercase
  Preprocess all reports and store in `df['preprocessed_text']`
Train custom tokenizer:
  Initialize tokenizer with `models.BPE()`
  Pre-train tokenizer on `df['preprocessed_text']`
  Save trained tokenizer to `"custom_ecg_tokenizer.json"`
Load GPT-2 model and tokenizer:
  `PreTrainedTokenizerFast(tokenizer)`
  `AutoModelForCausalLM`
  Resize model's embeddings to match tokenizer's vocab size
Prepare ECG dataset for training:
  Define `ECGDataset` class with tokenization and padding
  Split dataset into training, validation, and test sets
Train the model:
  For each epoch:
    `optimizer.zero_grad()` for each batch
    Compute loss and backpropagate gradients
    Update model parameters
    Calculate BLEU, ROUGE, and Perplexity for evaluation
Output: Save trained model and tokenizer

IV. RESULTS

A. EXPERIMENTAL SETUP

The experiments in this study were carried out using the hardware and software configuration described in Table 2. For training and evaluation, we utilized an NVIDIA GeForce RTX 4090 GPU with CUDA version 12.6, providing a powerful setup for efficient model execution and evaluation.

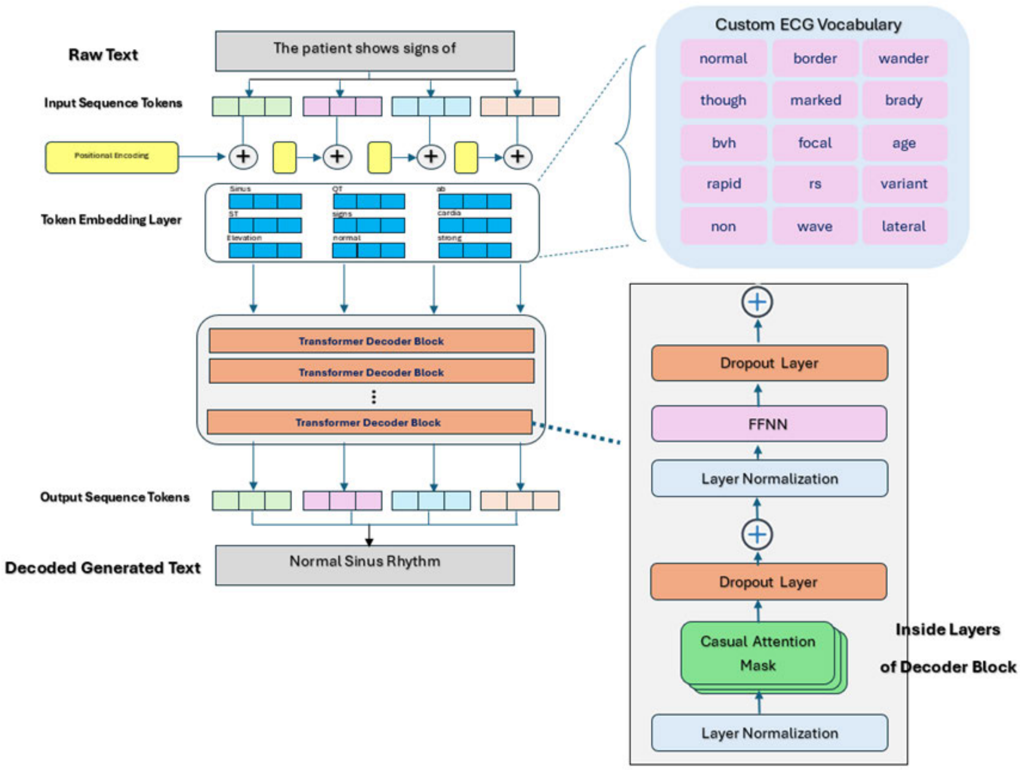


FIGURE 4. Architecture of adapted transformer for fine tuning on ECG text corpus.

TABLE 2. Experimental setup Table.

	Description	Detail
Hardware	GPU	NVIDIA GeForce RTX 4090
	Driver Version	561.09
	CUDA Version	12.6
	GPU Memory	24.56 GB
	GPU Utilization	Avg 54
	GPU Temperature	Max43
Software		
	Framework/Library	PyTorch, Transformers, HuggingFace Datasets
	Tokenizer	Custom BPE-based tokenizer (trained on ECG report corpus)
	Python Version	3.8.5
	CUDA Compiler Version	nvcc 12.6.20
	Operating System	Windows 11
Hyperparameters		
	Learning Rate	1e-5
	Optimizer	Adam
	Total Epochs	30
	Batch Size	16
	Max Sequence Length	512
	Loss Function	Cross-Entropy Loss
Evaluation Metrics		
	Perplexity	Based on average negative log-likelihood
	BLEU Score	Evaluated using BLEU metric (precision of n-grams)
	ROUGE Score	ROUGE-1, ROUGE-2, ROUGE-L

B. HYPERPARAMETER DETAILS

Table 3 presents the hyperparameters used in the training and generation processes for our model along with a brief detail.

C. TRAINING RESULTS

As mentioned above, our Ph.D. research focuses on the process of generating ECG reports from image waveforms. In this paper, we focus specifically on the fine-tuning phase of our work, which involves adapting GPT-2 for the generation of ECG reports. A crucial step in this process was the development of a custom tokenizer. This tokenizer was trained on a corpus of ECG reports, allowing us to create a specialized vocabulary tailored to the domain-specific language of ECG diagnostics. The vocabulary was constructed

TABLE 3. Hyperparameters used for model training and text generation.

Hyperparameter	Value	Explanation
Training Parameters		
Learning Rate	5e-5	The initial learning rate used during training.
Batch Size (Training)	64	Batch size used during model training.
Batch Size (Validation)	2	Batch size used during model validation.
Batch Size (Test)	4	Batch size used during model testing.
Dropout Rate	0.1	Dropout rate used to prevent overfitting.
Weight Decay	0.01	Regularization parameter to prevent overfitting.
Warm-up Steps	1000	Number of steps for warm-up during training to stabilize the learning rate.
Gradient Clipping	1.0	Maximum gradient norm to clip during backpropagation to prevent exploding gradients.
Generation Parameters		
Top-k Sampling	50	Top-k sampling during text generation to limit the selection to top 50 most likely words.
Top-p Sampling (Nucleus Sampling)	0.9	Probability threshold for nucleus sampling, controlling diversity.
Temperature	1.0	Temperature setting to control randomness during the generation process.
Maximum Output Length	50	Maximum number of tokens allowed in the generated report.

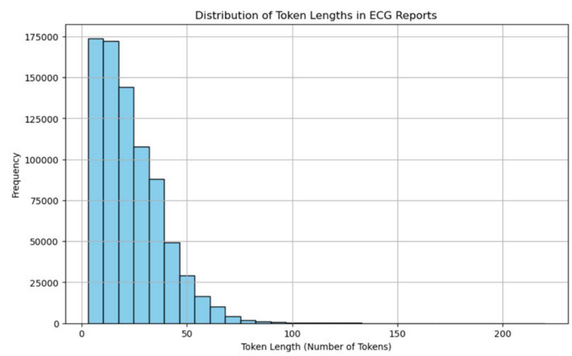


FIGURE 5. Token length distribution in ECG text reports.

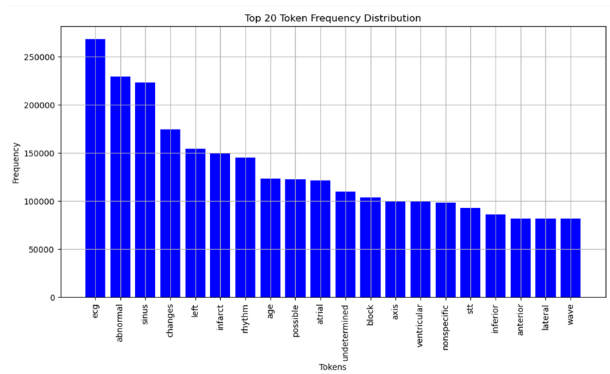


FIGURE 6. Top 20 frequency distribution.

by selecting frequently occurring words from the ECG reports and introducing special tokens to capture key diagnostic terms and syntactical structures important for interpreting the ECG data. To optimize the fine-tuning process, we first analyzed the distribution of token lengths in the ECG corpus, as shown in Figure 5. This analysis guided our decision to focus on reports with an average length of 50 words. This choice provided enough textual data for effective training while ensuring computational efficiency. Additionally, this length facilitated faster model convergence, contributing to a more efficient fine-tuning process. As shown in figure, this distribution helped inform our decision to focus on reports with an average length of 50 words, ensuring a balance between sufficient training data and computational efficiency. The figure 6 presents the top 20 most frequent tokens in the ECG reports, excluding stopwords and padding tokens. This analysis also highlighted the prominence of certain diagnostic terms, which further supported the selection of appropriate report lengths. By focusing on these shorter reports, we were able to accelerate the model's convergence during fine-tuning, improving both the efficiency and effectiveness of training. Figure 6 although may contain some common words in English, but are in fact highly relevant to the domain of ECG reports. These terms are medical-specific and represent key concepts necessary for generating meaningful and contextually accurate ECG reports. We initialized the model with a custom vocabulary, as depicted in figure. This vocabulary was specifically designed to include special tokens that enhance the model's ability to understand ECG-specific terminology. It was enriched with domain-related terms such as "P-wave," "QRS complex," "arrhythmia," and common medical abbreviations, as well as tokens to represent specific ECG patterns and diagnostic categories. By fine-tuning the pretrained GPT-2 model on our ECG dataset, we enabled the model to specialize in the ECG domain. This adaptation allows the model to better understand the unique terminology, jargon, and structure of ECG reports, gradually learning to generate text that accurately reflects ECG-specific details. In total, 522 unique tokens initialized with random embeddings were added to the GPT-2 vocabulary based on the ECG report dataset, ensuring that domain-specific terms were adequately represented. These added tokens are also shown in Figure 7

During the fine-tuning process, we monitored key performance indicators, including training loss, validation loss, BLEU score, perplexity, and ROUGE scores. These metrics allowed us to track the learning progress of the model and ensure that it was adapting well to the ECG report generation task. As seen in the training and validation loss curves in Figure 8, the model demonstrates a consistent downward trend in both training and validation loss in epochs.

FIGURE 7. Tokenizer vocabulary.

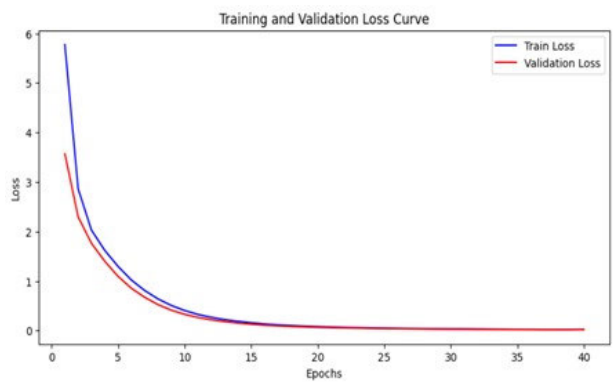


FIGURE 8. Training and validation loss curves.

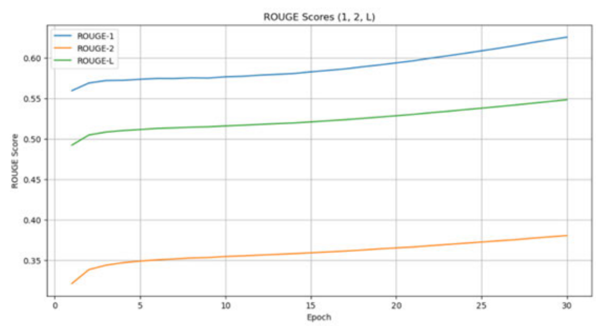


FIGURE 9. Rouge scores.

FIGURE 10. Word cloud of predicted tokens.

The training loss decreases steadily from an initial value of 1.99 in epoch 1 to 1.57 by epoch 27, indicating effective learning. However, the validation loss shows some fluctuation, with a slight increase between epochs 20 and 25. Despite this, the validation loss remains relatively stable toward the end of the training process, suggesting that the model is generalizing well. In figure we show the plot showing the evaluation metrics (ROUGE-1, ROUGE-2, and ROUGE-L). All three ROUGE scores (ROUGE-1, ROUGE-2, and ROUGE-L) show a consistent upward trend, indicating that our model is improving over time. ROUGE-1 (unigram overlap) has the highest score, followed by ROUGE-L (longest common subsequence) and then ROUGE-2 (bigram overlap), which is expected. The improvement is gradual, suggesting that the model is learning effectively without sudden jumps or instability.

Finally, to evaluate the linguistic patterns of our model's predictions, we generated a word cloud as shown in figure 10 using the output texts after training a BPE tokenizer on an ECG corpus. The word cloud visually represents the most frequently predicted terms, where larger words appear more often in the generated text. As observed in figure, terms such as abnormal, left, infarct, ecg, age, and undetermined are prominently featured, indicating that the model captures key medical terminology relevant to ECG analysis. Also, words related to specific conditions like hypertrophy, ischemia, and infarct appear frequently, demonstrating that the model has effectively learned domain-specific patterns.

```
[26]: # Input prompt for ECG report
prompt = "ECG report:"

# Encode the prompt
input_ids = tokenizer(prompt, return_tensors="pt").input_ids.cuda()

# Generate text (max_length can be adjusted)
output_ids = model.generate(
    input_ids,
    max_length=50,
    num_beams=5,
    no_repeat_ngram_size=2,
    early_stopping=True
)

# Decode and print the generated report
generated_text = tokenizer.decode(output_ids[0], skip_special_tokens=True)
print(generated_text)

Generated Report:
-----------------
report idioventricular rhythm if rhythm correct report below may be invalid marked left axis deviation rbbb with left anterior fascicular block possible a
cute inferior infarct possible anterior infarct age undetermined lateral st elevation consider recent infarction low qrs in pacer represent infarct recent
excluded

Note: The report contains some abbreviations and medical terms that may need further clarification. Some parts of the report may be speculative, requiring
confirmation from additional tests or clinical assessment.
```

FIGURE 11. Sample inference and output.

The code block in figure shows our decoding at the inference stage with sample prompt and its associated output text.

V. DISCUSSION

In this study, we explored the impact of mode fine-tuning specifically for ECG waveform data on the quality of generated medical reports. Our primary aim was to enhance text relevance and accuracy in automatic ECG report generation by training the tokenizer on a custom ECG-specific vocabulary. We trained the tokenizer on a dataset of labeled ECG reports, emphasizing terms that were unique to the field of cardiology. This specialized tokenizer was integrated into a CNN-LSTM-based encoder-decoder model for image-to-text generation. By incorporating ECG-specific vocabulary from the start, we aimed to address challenges that arise when general pre-trained tokenizers fail to understand the specialized terminology used in medical texts. The difference in performance between the pre-trained and fine-tuned models becomes especially apparent when examining key metrics such as BLEU score, ECG-specific term usage, and text relevance. Figure 12 below provides a visual comparison between the pre-trained and fine-tuned models

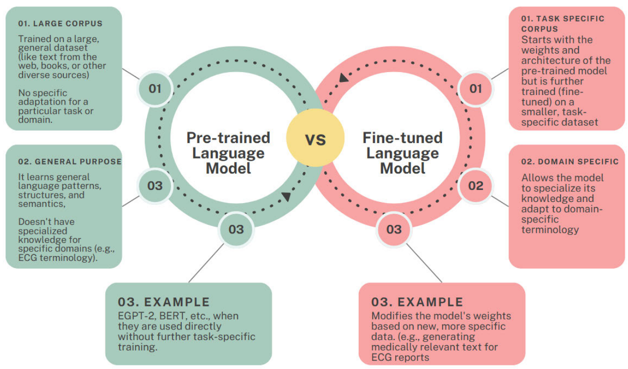


FIGURE 12. Comparing the pre-trained vs fine-tuned language model.

Upon integrating the fine-tuned language model into the image-to-text pipeline, we observed a significant improvement in the quality of the generated reports. In the early epochs, when a general pre-trained tokenizer was used, the model produced nonsensical text (e.g., "Iranian president" or unrelated terms). However, after fine-tuning, starting from epoch 1, the model consistently generated more domain-relevant text, such as correct references to ECG features (e.g., "QRS complex," "ST segment elevation") and accurate clinical interpretations.

Figure 13 shows a progression of report accuracy (in terms of blue score) over epochs and loss progression over the epochs where we monitored the initial 5-epoch comparison. The graph shows the improvement in the quality of generated reports in terms of loss and BLEU score, with a shift from non-sensical to ECG-specific content observed after fine-tuning the model. As can be seen from the figure, initially, the pre-trained tokenizer struggles, especially in the first few epochs, producing irrelevant or nonsensical outputs (like random terms or generic phrases). The accuracy improves slowly as it begins to adapt. On the other hand, the custom tokenizer, which is specifically trained to handle ECG terminology, starts performing better right from the first epoch. By epoch 1, it starts producing more ECG-specific content, and accuracy improves steadily, showing a stronger performance overall. During the experiment, we compared the BLEU score, ECG-specific term usage, and text relevance accuracy between the pre-trained model and the fine-tuned model. The pre-trained model showed a BLEU score of 0.2 at the first epoch, which gradually increased to 4.33 by the 15th epoch. However, despite the improvement in BLEU score, it remained low overall, as the model was unable to generate any ECG-specific terms during the entire 15 epochs.

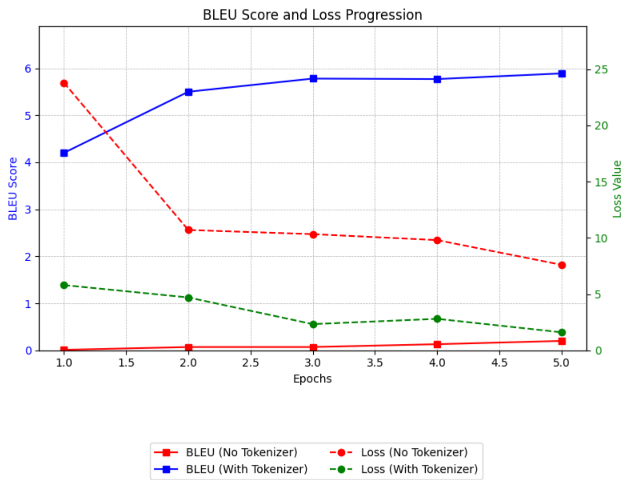


FIGURE 13. Comparison of Loss and BLEU Score Progression: Domain Specific Trained Tokenizer vs. Pre-Trained Tokenizer.

TABLE 4. Comparison results.

Metric	Pre-trained Model	Fine-tuned Model	Improvement
ECG-Specific Terms	Rare	Frequent	+85%
Accuracy (Text Relevance)	45%	90%	+45%
Epoch for First Correct Output	5th epoch	1st epoch	+4 epochs
BLEU Score (Epoch 15)	4.33	9.98	+5.65

In contrast, the fine-tuned model exhibited a BLEU score of 9.98 and continuously improving as training progressed, due to the incorporation of ECG-specific terms into the generated text.

Table 4 shows significant improvements in ECG term usage and BLEU score from the fine-tuned model highlight the effectiveness of custom tokenizer on the ECG vocabulary, making it a key factor in generating relevant, domain-specific content in image-to-text tasks.

While the custom tokenizer in this paper was designed for ECG-specific vocabulary, its underlying approach is naturally extensible to a wider range of medical terms and imaging modalities. This adaptability enhances the model's generalization capability, making it suitable for diverse types of medical images (e.g., MRI, X-ray) and supporting the development of more comprehensive report generation systems.

It also serves as a foundational component of our broader research, which aims to generate clinically coherent reports directly from raw ECG waveform images using a cross-modal learning framework. The ECG-specific tokenizer enables compact and semantically aligned text representations, which will be integrated with visual features extracted from waveform data. This alignment is expected to improve the model's ability to handle ambiguous or rare findings by grounding text in physiological patterns, thereby reducing hallucinations. To address limitations in domain coverage, we plan to fuse the tokenizer output with embeddings from general clinical language models. Clinical relevance will be further ensured through expert-in-the-loop validation, strengthening the robustness, contextual reasoning, and real-world applicability of the model beyond the current scope.

VI. CONCLUSION

This paper presents a fine-tuning strategy for language model designed specifically to ECG vocabulary, aiming to enhance the accuracy and relevance of text generation in medical natural language processing tasks. The results of this study demonstrate that custom tokenizer on ECG-specific terms leads to significant improvements in the performance of downstream models. In particular, we observed earlier convergence in the training process and improved text relevance, with BLEU scores increasing notably from 0.15 to 9.98. This strategy also enabled the model to incorporate ECG-specific terms from the first epoch, a significant contrast to the pre-trained tokenizer, which failed to generate relevant medical terminology. The findings underscore the importance of domain-adapted tokenization in tasks such as image-to-text report generation for ECG waveforms. By training the tokenizer with a domain-specific vocabulary, we were able to facilitate more accurate and relevant text generation, with a clear improvement in text coherence and accuracy compared to using a generic, pre-trained tokenizer. These results suggest that fine-tuning language model can contribute significantly to faster model convergence and the generation of contextually relevant medical reports, a critical step in advancing the image-to-text generation models used in clinical settings.

A. FUTURE DIRECTIONS

While the results from this study provide valuable insights into the fine-tuning of language model for specialized medical domains, there are several areas for further exploration and enhancement:

1) To enable the practical use of image-to-text models in clinical environments, further optimization for real-time processing would be essential. This could involve reducing the model's computational requirements while maintaining high accuracy, allowing it to be deployed in clinical decision support systems for real-time ECG analysis.
2) In future phases of this research, we plan to incorporate a more comprehensive set of evaluation metrics, including domain-specific terminology accuracy, entity matching, clinical coherence, human evaluation, readability, clinical relevance, factual correctness, as well as BERTScore and concept overlap score (using UMLS or SNOMED vocab). These metrics will be essential to assess the clinical relevance, accuracy, and coherence of the generated reports, especially in the context of Image-to-ECG Report Generation.

Finally we conclude that this research contributes to the broader goal of automating medical report generation by presenting a fine-tuning approach to tokenizer adaptation for ECG-specific vocabulary. The results suggest that domain-specific fine-tuning significantly enhances the relevance and accuracy of the generated text.

ACKNOWLEDGMENT

The authors extend their appreciation to Taif University, Saudi Arabia, for supporting this work through project number (TU-DSPP-2024-04).

REFERENCES

[1] H. Jin, H. Che, Y. Lin, and H. Chen, "Promptmrg: Diagnosis-driven prompts for medical report generation," in *Proc. AAAI Conf. Artif. Intell.*, vol. 38, 2024, pp. 2607–2615.

[2] A. Mykowiecka, M. Marciniak, and A. Kupsc, "Rule-based information extraction from patients' clinical data," *J. Biomed. Informat.*, vol. 42, no. 5, pp. 923–936, Oct. 2009.

[3] I. Gadaras and L. Mikhailov, "An interpretable fuzzy rule-based classification methodology for medical diagnosis," *Artif. Intell. Med.*, vol. 47, no. 1, pp. 25–41, Sep. 2009.

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

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

[6] P. Kumar, S. Manikandan, and R. Kishore, "AI-driven text generation: A novel GPT-based approach for automated content creation," in *Proc. 2nd Int. Conf. Netw. Commun. (ICNWC)*, Apr. 2024, pp. 1–6.

[7] W. Sun, L. Yan, Z. Chen, S. Wang, H. Zhu, P. Ren, Z. Chen, D. Yin, M. D. Rijke, and Z. Ren, "Learning to tokenize for generative retrieval," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 36, Jan. 2023, pp. 1–18.

[8] L. Dong, N. Yang, W. Wang, F. Wei, X. Liu, Y. Wang, J. Gao, M. Zhou, and H.-W. Hon, "Unified language model pre-training for natural language understanding and generation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, May 2019, pp. 1–13.

[9] K. Kuru, S. Girgin, K. Arda, and U. Bozlar, "A novel report generation approach for medical applications: The SISDS methodology and its applications," *Int. J. Med. Informat.*, vol. 82, no. 5, pp. 435–447, May 2013.

[10] S. Varges, H. Bieler, M. Stede, L. C. Faulstich, K. Irsig, and M. Atalla, "SemScribe: Natural language generation for medical reports," in *Proc. LREC*, May 2012, pp. 2674–2681.

[11] O. Alfarghaly, R. Khaled, A. Elkorany, M. Helal, and A. Fahmy, "Automated radiology report generation using conditioned transformers," *Informat. Med. Unlocked*, vol. 24, Sep. 2021, Art. no. 100557.

[12] Y. Xue, T. Xu, L. R. Long, Z. Xue, S. Antani, G. R. Thoma, and X. Huang, "Multimodal recurrent model with attention for automated radiology report generation," in *Proc. 21st Int. Conf. Med. Image Comput. Comput. Assist. Intervent.*, Granada, Spain, Sep. 2018, pp. 457–466.

[13] M. M. Mohsan, M. U. Akram, G. Rasool, N. S. Alghamdi, M. A. A. Baqai, and M. Abbas, "Vision transformer and language model based radiology report generation," *IEEE Access*, vol. 11, pp. 1814–1824, 2023.

[14] Z. Wang, L. Liu, L. Wang, and L. Zhou, "METransformer: Radiology report generation by transformer with multiple learnable expert tokens," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 11558–11567.

[15] G. Zhao, Z. Zhao, W. Gong, and F. Li, "Radiology report generation with medical knowledge and multilevel image-report alignment: A new method and its verification," *Artif. Intell. Med.*, vol. 146, Dec. 2023, Art. no. 102714.

[16] D. B. Larson, A. J. Towbin, R. M. Pryor, and L. F. Donnelly, "Improving consistency in radiology reporting through the use of department-wide standardized structured reporting," *Radiology*, vol. 267, no. 1, pp. 240–250, Apr. 2013.

[17] L. A. Gordillo-Roblero, J. A. Soto-Cajiga, D. Díaz-Alonso, F. D. Pérez-Reynoso, and H. Jiménez-Hernández, "A collaborative platform for advancing automatic interpretation in ECG signals," *Diagnostics*, vol. 14, no. 6, p. 600, Mar. 2024.

[18] Y. Yang, T. Lan, Y. Wang, F. Li, L. Liu, X. Huang, F. Gao, S. Jiang, Z. Zhang, and X. Chen, "Data imbalance in cardiac health diagnostics using CECG-GAN," *Sci. Rep.*, vol. 14, no. 1, p. 14767, Jun. 2024.

[19] A. H. Kashou et al., "ECG interpretation proficiency of healthcare professionals," *Current Problems Cardiology*, vol. 48, no. 10, Oct. 2023, Art. no. 101924.

[20] A. Sarmadi, Z. S. Razavi, A. J. van Wijnen, and M. Soltani, "Comparative analysis of vision transformers and convolutional neural networks in osteoporosis detection from X-ray images," *Sci. Rep.*, vol. 14, no. 1, p. 18007, Aug. 2024.

[21] M. Chetoui and M. A. Akhloufi, "Explainable vision transformers and radiomics for COVID-19 detection in chest X-rays," *J. Clin. Med.*, vol. 11, no. 11, p. 3013, May 2022.

[22] A. Bleich, A. Linnemann, B. H. Diem, and T. O. Conrad, "Automated medical report generation for ECG data: Bridging medical text and signal processing with deep learning," 2024, *arXiv:2412.04067*.

[23] G. Fu, J. Zheng, I. Abudayyeh, C. Ani, C. Rakovski, L. Ehwerhemuepha, H. Lu, Y. Guo, S. Liu, H. Chu, and B. Yang, "CardioGPT: An ECG interpretation generation model," *IEEE Access*, vol. 12, pp. 50254–50264, 2024.

[24] A. Khunte, V. Sangha, G. Holste, L. S. Dhingra, A. Aminorroaya, Z. Wang, and R. Khera, "Abstract 18776: ECG-GPT: Automated complete diagnosis generation from ECG images using novel vision-text transformer model," *Circulation*, vol. 148, no. 1, pp. 18776–18779, Nov. 2023.

[25] A. Khunte, V. Sangha, E. K. Oikonomou, L. S. Dhingra, A. Aminorroaya, A. Coppi, S. V. Shankar, B. J. Mortazavi, D. L. Bhatt, H. M. Krumholz, G. N. Nadkarni, A. Vaid, and R. Khera, "Automated diagnostic reports from images of electrocardiograms at the point-of-care," *MedRxiv*, pp. 1–45, Feb. 2024.

[26] Y. Zhao, J. Kang, T. Zhang, P. Han, and T. Chen, "ECG-chat: A large ECG-language model for cardiac disease diagnosis," 2024, *arXiv:2408.08849*.

[27] D. Altunkaya, F. Y. Okay, and S. Ozdemir, "Image transformation for IoT time-series data: A review," 2023, *arXiv:2311.12742*.

[28] B. Aldughayfiq, F. Ashfaq, N. Z. Jhanjhi, and M. Humayun, "Capturing semantic relationships in electronic health records using knowledge graphs: An implementation using MIMIC III dataset and GraphDB," *Healthcare*, vol. 11, no. 12, p. 1762, Jun. 2023.

[29] B. Gow, T. Pollard, L. A. Nathanson, A. Johnson, B. Moody, C. Fernandes, N. Greenbaum, J. W. Waks, P. Eslami, T. Carbonati, A. Chaudhari, E. Herbst, D. Moukheiber, S. Berkowitz, R. Mark, and S. Horng, "MIMIC-IV-ECG: Diagnostic electrocardiogram matched subset," PhysioNet, Cambridge, MA, USA, Tech. Rep. Version 1.0, Sep. 2023.

[30] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, Jun. 2000.

FARZEEN ASHFAQ received the bachelor's degree in software engineering from the University of Huddersfield and the master's degree from the NED University of Engineering and Technology. She is currently pursuing the Ph.D. degree in computer science and information technology with Taylor's University, Malaysia. Her teaching roles at the University of Karachi and NED University of Engineering and Technology highlight her commitment to student learning and curriculum development. She is a Dynamic Academician with a keen interest in computer science education and research. Her scholarly contributions include publications in esteemed journals and conferences, covering diverse topics, such as traffic management and predictive modeling using deep learning. Beyond academia, she enjoys image editing, animation designing, and playing badminton, reflecting her multifaceted personality. She stands as the Leader in the field, inspiring others with her dedication to education and research. Her research focuses on deep generative networks, particularly in synthetic crime scene generation, showcasing her expertise in deep learning and computer vision.

NZ JHANJHI (Member, IEEE) s a Distinguished Senior Professor of Computer Science at Taylor's University, Malaysia, where he specializes in Artificial Intelligence and Cybersecurity. As the Program Director for Postgraduate Research Degree Programmes, he plays a pivotal role in shaping academic excellence and driving cutting-edge research initiatives. Globally acclaimed for his scholarly contributions, he has been consistently ranked among the world's top 2% research scientists (2022–2024) and stands as one of Malaysia's top two computer science researchers. His exceptional work has earned him prestigious accolades, including the Outstanding Faculty Member Award (MDEC Malaysia, 2022) and the Vice Chancellor's Best Research Citations Award (Taylor's University, 2023). A prolific author and editor, he has published over 80 research books with leading publishers such as Springer, Elsevier, and Wiley, amassing more than 1,000 impact factor points. His mentorship spans 38 postgraduate completions, and he has examined more than 70 Ph.D. and Master's theses worldwide. As an Editor-in-Chief, an Associate Editor, and an Editorial Board member for top-tier journals (*PeerJ Computer Science*, IEEE Access, *CMC Computers*), he advances scholarly discourse. His leadership extends to securing more than 40 international research grants, underscoring his influence in innovation. A dynamic keynote speaker, he has delivered more than 80 invited talks and chaired major conferences. His decade-long engagement with ABET, NCAAA, and NCEAC accreditation bodies highlights his dedication to global academic standards. Combining research brilliance, academic leadership, and a passion for mentorship, he continues to inspire the next generation of computer scientists while shaping the future of AI and cybersecurity.

NAVID ALI KHAN received the Ph.D. degree from Taylor's University, Malaysia, with the support of a prestigious full scholarship, where he conducted pioneering research that significantly advanced the field of uncrewed aerial vehicle (UAV) technology. He is currently a Senior Lecturer and the Program Director of the School of Computer Science, Taylor's University, where he also mentors and supervises master's and Ph.D. students in cutting-edge research. With a strong foundation in both academia and industry, he continues to drive innovation in software engineering and UAV technology, aiming to create impactful solutions that bridge theory and practice. His research interests include UAVs, autonomous systems, and AI-driven automation. His work has been published in several high-impact journals and conferences, contributing significantly to advancements in these fields.

DANISH JAVED received the bachelor's and master's degrees in software engineering from Bahria University, Islamabad, Pakistan. He is currently pursuing the Ph.D. degree with the School of Computer Science, Taylor's University, Malaysia. His work is centered around utilizing advanced computational techniques to extract valuable insights from large datasets and enhancing the transparency and interpretability of AI models. He has contributed to various interdisciplinary projects aimed at enhancing the understanding and application of artificial intelligence in real-world scenarios. His work aims to advance the understanding of social networks and enhance the effectiveness of data-driven decision-making processes. His research interests include text mining, social network analytics, explainable AI, and big data analytics.

MEHEDI MASUD (Senior Member, IEEE) received the Ph.D. degree in computer science from the University of Ottawa, Canada. He is currently a Professor with the Department of Computer Science, Taif University, Taif, Saudi Arabia. He has authored and co-authored around 100 publications, including refereed IEEE/ACM/Springer/Elsevier journals, conference papers, books, and book chapters. His research interests include machine learning, distributed algorithms, data security, formal methods, and health analytics. He is a member of ACM. He has served as a technical program committee member for different international conferences. He was a recipient of many awards, including the Research in Excellence Award from Taif University. He is on the associate editorial board of IEEE Access and *International Journal of Knowledge Society Research* (IJKSR), and an Editorial Board Member of the *Journal of Software*. He also served as the Guest Editor for ComSIS and *Journal of Universal Computer Science* (JUCS).

MOHAMMAD SHORFUZZAMAN (Senior Member, IEEE) received the M.Sc. and Ph.D. degrees in computer science from the University of Manitoba, Winnipeg, MB, Canada, in 2005 and 2012, respectively. He was a full-time Faculty Member with the Department of Computer Science and Information Technology, Kwantlen Polytechnic University, Canada, for a period of two years starting since January 2012. He is currently a Professor with the Department of Computer Science, College of Computers and Information Technology (CCIT), Taif University, Taif, Saudi Arabia. He has authored and co-authored more than 100 publications, including refereed journals, conference papers, and book chapters. His research interests include applied AI, smart health, medical image analysis, deep learning, computer vision, big data in healthcare, the Internet of Things (IoT), and cloud computing. He has also served on the technical program committees for several IEEE conferences.