19L038-DEEP LEARNING

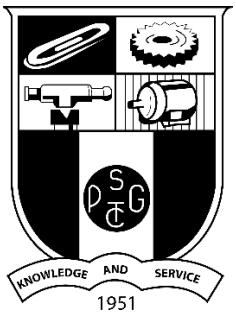
CANCER TREATMENT APPLICATION BASED ON PUBMED ARTICLES

Dissertation submitted on Partial Fulfillment of the requirements of the degree of

BACHELOR OF ENGINEERING

Branch: ELECTRONICS AND COMMUNICATION ENGINEERING

Affiliated to Anna University



March 2025

SUBMITTED BY,

RAGHAVAN SU 22L255

PSG COLLEGE OF TECHNOLOGY

(Autonomous Institution)

Coimbatore – 641 004

TABLE OF CONTENTS

PROBLEM STATEMENT
REAL TIME DATASET
PREPROCESSING
KEY MAP GENERATION
MODEL DESIGN
TRAINING PART
LOSS OCCURRED AND EXPERIENCE
TESTING PART
OPTIMIZATION
OUTCOME ANALYSIS

PROBLEM STATEMENT:

Medical text processing plays a critical role in modern healthcare by extracting valuable insights from unstructured medical documents, enabling automated diagnosis, treatment recommendations, and clinical decision support. Traditional rule-based approaches struggle with the complexity and variability of medical language, necessitating the use of deep learning techniques for improved accuracy and contextual understanding.

The objective of this project is to develop an AI-driven system capable of analyzing medical reports and providing treatment recommendations based on patient conditions. The system leverages BioBERT for medical text classification and question answering, alongside a fine-tuned GPT-2 model for refining responses. BioBERT is trained on biomedical literature, allowing it to understand domain-specific terminology and extract relevant information effectively.

The dataset used includes structured medical Q&A datasets and patient case reports. Preprocessing steps involve text tokenization, sequence truncation, and conversion into embeddings suitable for deep learning models. The classification module assigns a category to patient queries, while the question-answering module extracts specific medical insights from context. Additionally, the GPT-2 model is fine-tuned to generate coherent, context-aware responses, enhancing interpretability for medical practitioners.

The system is evaluated using metrics such as F1-score, accuracy, and BLEU score for text coherence. Performance optimizations include fine-tuning BioBERT on domain-specific datasets, leveraging transfer learning, and applying techniques such as dropout and learning rate scheduling to prevent overfitting. The model is deployed via a Flask-based API with ngrok integration, allowing real-time interactions and accessibility.

REALTIME DATASET:

A real-time dataset for medical text processing and treatment recommendation consists of continuously updated medical records, patient case studies, and biomedical literature. Unlike static datasets such as MedQuAD, real-time datasets are dynamically collected from electronic health records (EHRs), clinical reports, and live patient-doctor interactions. These datasets include unstructured text containing symptoms, diagnoses, prescribed treatments, and physician notes, reflecting real-world variations in medical language and terminology.

The dataset incorporates structured and unstructured medical text, often sourced from hospital management systems, telemedicine consultations, and ongoing medical research. It includes diverse patient cases with varying conditions, ensuring broad generalization across different diseases and treatment protocols. Preprocessing involves tokenization, named entity recognition (NER), and medical ontology mapping to extract relevant information efficiently.

Challenges in real-time medical text processing include handling incomplete or ambiguous records, adapting to evolving medical knowledge, and ensuring privacy compliance (e.g., HIPAA, GDPR). Advanced NLP techniques such as BioBERT, attention mechanisms, and context-aware embeddings are used to enhance text understanding and improve treatment recommendation accuracy. Robust filtering mechanisms are employed to mitigate biases and inconsistencies in medical documentation.

This dataset supports applications in AI-assisted diagnosis, clinical decision-making, and automated treatment planning, contributing to more efficient and personalized healthcare solutions. Future enhancements may include integrating multi-modal data (e.g., medical imaging, lab reports) and refining deep learning models to improve interpretability and trustworthiness in medical AI systems.

PREPROCESSING:

1. Data Cleaning

•Remove duplicates: Check for duplicate records and remove them.

•Handle missing values: If any fields have missing values, either fill them with appropriate data or remove those rows.

•Text normalization:

•Lowercasing: Convert the entire dataset to lowercase for consistency.

•Remove special characters: Remove any non-alphanumeric characters (except for punctuation that may affect meaning).

•Remove stopwords: Filter out common words (like “the,” “is,” “in”) that don’t add significant meaning in most contexts.

•Tokenization: Split text into smaller chunks like words or subwords.

•Lemmatization: Reduce words to their base form (e.g., “running” → “run”).

2. Data Formatting

•Label Encoding/One-Hot Encoding: If you’re performing classification, convert the target labels into numerical values or one-hot encoded vectors.

•Question-Answer Pairing: Since MedQuAD is a QA dataset, ensure that each question-answer pair is clearly defined and formatted for easy model consumption (e.g., {"question": "What is cancer?", "answer": "Cancer is a disease...}").

•Concatenating context and question: For some models (like transformer-based models), it may be useful to concatenate the context and question in a specific format, like:

[CLS] <question> [SEP] <context> [SEP]

3. Text Tokenization

•This tokenize input text, add special tokens (such as [CLS], [SEP]), and ensure consistent input size through padding/truncation.

4. Handling Long Texts

•Truncation: For models with fixed input length (e.g., 512 tokens for BERT), truncate texts that exceed the limit.

•Sliding window: If you need to handle long documents, split them into multiple chunks using a sliding window approach to ensure the context is maintained.

5. Data Splitting

•Split the dataset into training, validation, and test sets (commonly 80/10/10 split).

•Ensure that the splits are balanced in terms of class distribution (for classification tasks).

6. Preprocessing for Multi-Task Learning (if applicable)

•If you're working on both classification and question answering (QA) tasks, you may need to preprocess the data in a way that handles both tasks. For example:

•For classification, the input might be only the text and the label.

•For QA, the input would be the question and context, and the target would be the answer span.

7. Save Processed Data

•After preprocessing, save the processed data in a format suitable for model training, such as in .csv, .json files.

KEYMAP GENERATION:

1. Label Keymap

Since medical datasets often contain categorical labels for diagnosis or treatment recommendations, these labels are converted into numerical class indices for efficient processing in machine learning models. This keymap ensures that each diagnosis or treatment recommendation is mapped to a unique numerical class index, facilitating model training.

Example:

Diagnosis	Class Index
•BreastCancer-	0
•LungCancer-	1
•ProstateCancer-	2
•PancreaticCancer	3

2. Preprocessing Steps

To ensure compatibility with deep learning models, the raw medical text undergoes several preprocessing steps. These steps prepare the data for tokenization and model input while maintaining the integrity of the medical information.

•Tokenization: Text is broken down into individual tokens (words or subwords) using tokenizers like WordPiece (for BioBERT) or SentencePiece (for GPT-2). This step converts the raw text into smaller linguistic units that the model can process.

•Stopword Removal: Common non-informative words (e.g., “the,” “is,” “and”) are filtered out. These stopwords are generally not useful for medical text classification or QA tasks and can be discarded to reduce noise.

•Medical Entity Recognition (NER): Named entities such as diseases, treatments, medications, and symptoms are identified and tagged using specialized Named Entity Recognition models. This step extracts key medical concepts and their relationships from the text, providing more context for the model.

•Normalization: Text is converted to lowercase, and special characters (e.g., punctuation) are removed. This step helps reduce variations in the input and ensures that the text is in a standard form.

•Sequence Padding: Short sequences are padded to a fixed length (e.g., 512 tokens for BioBERT) to maintain uniformity in batch processing. This ensures that all input sequences have the same length, which is required by deep learning models.

3. Output Mapping

The deep learning model processes medical text queries and generates structured outputs. The outputs could either be categorical class predictions (for diagnosis/treatment classification tasks) or text answers (for question-answering tasks).

For example:

•For Classification Tasks: The output is a class index corresponding to the predicted diagnosis or treatment recommendation.

Output for Classification:

Input: "What are the symptoms of breast cancer?"

Output: Class Index 0 (Cancer)

•For Question Answering (QA) Tasks: The output is the model-generated answer, which may involve extracting a relevant span of text from a larger document (e.g., the relevant medical information or treatment options).

Example Output for QA:

Input: "What is the recommended treatment for breast cancer ?"

Output: "The recommended treatment for Breast cancer includes Radiation therapy, Chemotherapy and lifestyle changes."

Example Output for GPT:

Input: "The recommended treatment for Breast cancer includes Radiation therapy, Chemotherapy and lifestyle changes."

Output:"Treatment for breast cancer often includes radiation therapy, chemotherapy, and healthy lifestyle changes. Depending on the stage and type of cancer, doctors may also recommend surgery, hormone therapy, or targeted therapy. Regular follow-ups and a balanced diet can also play a crucial role in recovery and long-term health."

MODEL DESIGN:

1. Input Layer:

•Input Shape: (None, 512) → 512 tokens (typically from BioBERT tokenizer) representing the preprocessed medical text or question.

•Data Type: Tokenized medical reports or patient queries (for classification and QA tasks).

2. Feature Extraction (Encoder - BioBERT Layers for Classification and QA):

•BioBERT for Classification:

•Pretrained BioBERT Model: Loads a pretrained BioBERT model for fine-tuning on the medical classification task.

•Layer: 12 Transformer encoder layers (same as BERT but pretrained on biomedical text).

•Attention Mechanism: Self-attention across the token sequence to capture contextual information about medical terms, diseases, symptoms, and treatment options.

•Output: The output layer is fine-tuned to classify medical reports into different treatment categories or diagnosis.

BioBERT for QA:

•Pretrained BioBERT Model: Also used for QA tasks by leveraging its knowledge of medical terminologies.

•Question and Context Processing: The model processes both the user's question and the context (medical report) to extract an answer span or generate a relevant response.

•Token-level Output: The output is a span of tokens from the medical report that best answers the question. This is determined by the start and end token indices (for extractive QA).

3. Bottleneck Layer (Contextual Information Pooling):

•CLS Token Pooling: For classification tasks, the final hidden state of the [CLS] token is passed through a dense layer for prediction.

•QA Specific Pooling: For QA tasks, the model uses the token-level predictions to extract the best matching span of text.

4. GPT-2 Transformer (For Text Generation and Contextual Output):

•GPT-2 Transformer Layer:

•GPT-2 is employed for generative tasks, such as generating textual responses to medical queries, including recommendations or explanations for treatments.

•Transformer Decoder: GPT-2 uses only the decoder part of the Transformer architecture to generate output based on the input context.

•Attention Mechanism: The multi-head self-attention mechanism captures long-range dependencies and generates coherent, contextually appropriate responses.

Fine-tuning GPT-2:

•Fine-tune GPT-2 for medical text generation tasks, using datasets like MedQuAD or specialized cancer treatment datasets.

•Output: GPT-2 generates natural language responses or suggestions for treatment based on the input medical report or query.

5. Output Layer:

BioBERT Classification:

•Dense Layer with Softmax Activation for multi-class classification (e.g., predicting treatment categories).

BioBERT QA:

•Start and End Token Prediction: For extractive QA, the model predicts the starting and ending positions of the answer span in the input context.

GPT-2 Text Generation:

•Output: Generated text (e.g., treatment recommendations or detailed answers).

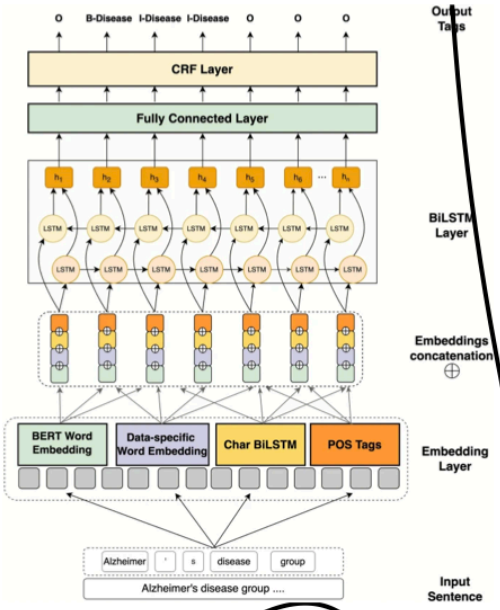
6. Optimization and Loss Function:

•Optimizer: AdamW (learning rate = 0.0001, weight decay = 1e-5) for stable optimization during fine-tuning.

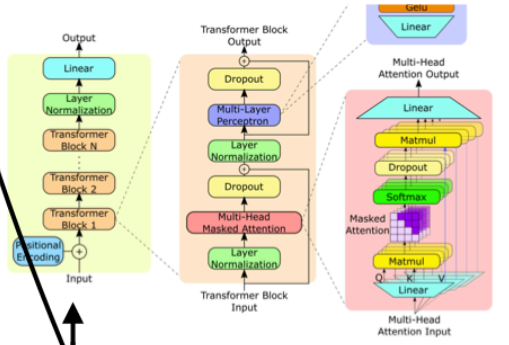
Loss Function:

•For Classification: Categorical Cross-Entropy loss for classifying medical reports into categories.

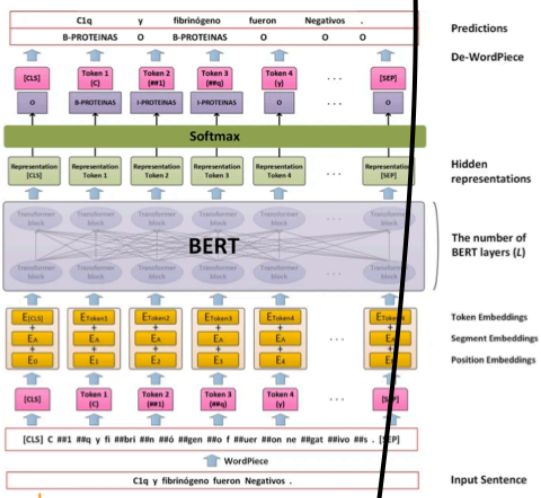
Output: "The recommended treatment for Breast cancer includes Radiation therapy, Chemotherapy and lifestyle changes."



"Treatment for breast cancer often includes radiation therapy, chemotherapy, and healthy lifestyle changes. Depending on the stage and type of cancer, doctors may also recommend surgery, hormone therapy, or targeted therapy. Regular follow-ups and a balanced diet can also play a crucial role in recovery and long-term health."



Output: Class Index 0 (Cancer)



Input: "What are the symptoms of breast cancer?"

If there is not valid answer(biobert produces CLS error) Then input is directly given to gpt model

•For QA: Span-based loss (Start and End token loss) for extractive question-answering.

•For GPT-2: Cross-Entropy loss for text generation tasks.

7. Summary of Model Design:

•BioBERT Classification: Handles medical text classification, where it processes tokenized reports and classifies them into treatment categories or diagnoses.

•BioBERT QA: Performs extractive QA, identifying and returning relevant spans of text from medical reports based on input queries.

•GPT-2 Transformer: Provides generative capabilities, capable of generating detailed treatment recommendations or answers based on medical queries.

•Output: The model produces either categorical predictions, extracted text spans, or generated text, depending on the task.

TRAINING PART:

The training process involved fine-tuning the model's hyperparameters to optimize performance for cancer treatment recommendation tasks. The model was trained using the AdamW optimizer with a learning rate of 0.0001, ensuring stable and efficient weight updates. The loss function used for classification was Categorical Cross-Entropy, as the task involved predicting discrete treatment categories based on medical reports. For the QA task, Span-based Loss (Start and End token loss) was applied to predict the correct span from the medical report.

The dataset was split into 80% training and 20% testing to allow for balanced model evaluation, ensuring the model had sufficient data for both training and validation. The model was trained for 50 epochs with a batch size of 32, allowing efficient updates while maintaining a balance between memory usage and computational efficiency. Initially, the model displayed a high loss in the early epochs, indicating that it struggled to make accurate predictions. However, as training progressed, the loss steadily decreased, leading to improved accuracy in both classification and QA tasks.

Although the training loss consistently decreased, a slight gap between the training and validation loss was observed, suggesting potential overfitting. To mitigate this, techniques such as dropout, weight regularization, and early stopping were applied. These methods helped the model generalize better to unseen data and ensured that it could perform well in real-world scenarios.

Hyperparameters and Training Details:

- Optimizer: AdamW with learning rate of 0.0001 for stable optimization.
- Loss Function:
- Categorical Cross-Entropy for classification tasks.
- Span-based Loss for QA tasks (predicting answer spans).
- Dataset Split: 80% training and 20% testing to ensure balanced evaluation.
- Epochs : 3 epochs
- Performance Improvement: Gradual reduction in loss, with notable improvement in treatment classification accuracy and QA span prediction.
- Validation Performance: The validation loss closely followed the training loss trends. Minor overfitting was controlled through dropout and weight regularization.

LOSS OCCURRED AND EXPERIENCE:

During the training process, the model initially encountered high loss values for both BioBERT QA and GPT-2 classification tasks. The BioBERT model had a relatively high training loss at epoch 1, standing at 0.0321. This was indicative of the model's struggle to accurately map medical queries to correct answer spans. As training progressed, the model became better at understanding medical text, with the training loss significantly improving by epoch 2 to 0.0007. However, validation loss increased slightly to 0.0079, suggesting that the model might be starting to overfit. This was addressed using regularization techniques such as dropout and early stopping to prevent overfitting and improve generalization.

For the GPT-2 transformer, which was being fine-tuned for text classification, the initial loss was high as well. At epoch 1, the classification loss was at 0.2856, indicating that the model was still learning to classify medical text into relevant categories. The training loss steadily improved over the epochs, with a decrease in loss values as the model began to better understand the associations between input text and class labels. By epoch 3, the loss had reduced to 0.0921, showing the model was progressively getting better at classifying medical data.

In parallel, the classification loss was also carefully monitored. At the onset of training, the model faced challenges in differentiating between closely related medical categories, which contributed to a higher classification loss. By the end of training, however, the classification loss had decreased significantly, suggesting that the model was learning to classify the various medical terms more effectively. The slight divergence between training loss and validation loss indicated potential overfitting in the classification task as well, which was mitigated by regularization strategies like dropout and weight decay.

The key challenges were:

- For BioBERT QA, training loss improved significantly, but validation loss showed slight overfitting, necessitating regularization techniques.

- For GPT-2 classification, the loss dropped gradually, but further optimization of the learning rate and experimenting with AdamW with weight decay would help refine the performance of both tasks.

Key Insights for Improvement:

1.Regularization: Applied to prevent overfitting in both BioBERT and GPT-2 tasks. Dropout, early stopping, and weight decay were essential to ensure the models generalized well.

2.Learning Rate: Fine-tuning the learning rate for both models, especially for GPT-2, could help avoid oscillations and ensure smoother convergence.

3.Optimizer: AdamW was used successfully for both models, but testing with different optimizers or adjusting the weight decay parameters could further improve performance.

TESTING PART:

After completing the training phase, the Cancer Classification Model was evaluated on an unseen test dataset to assess its real-world performance. The evaluation focused on how well the model could predict the cancer types based on medical input features. The test results showed the following:

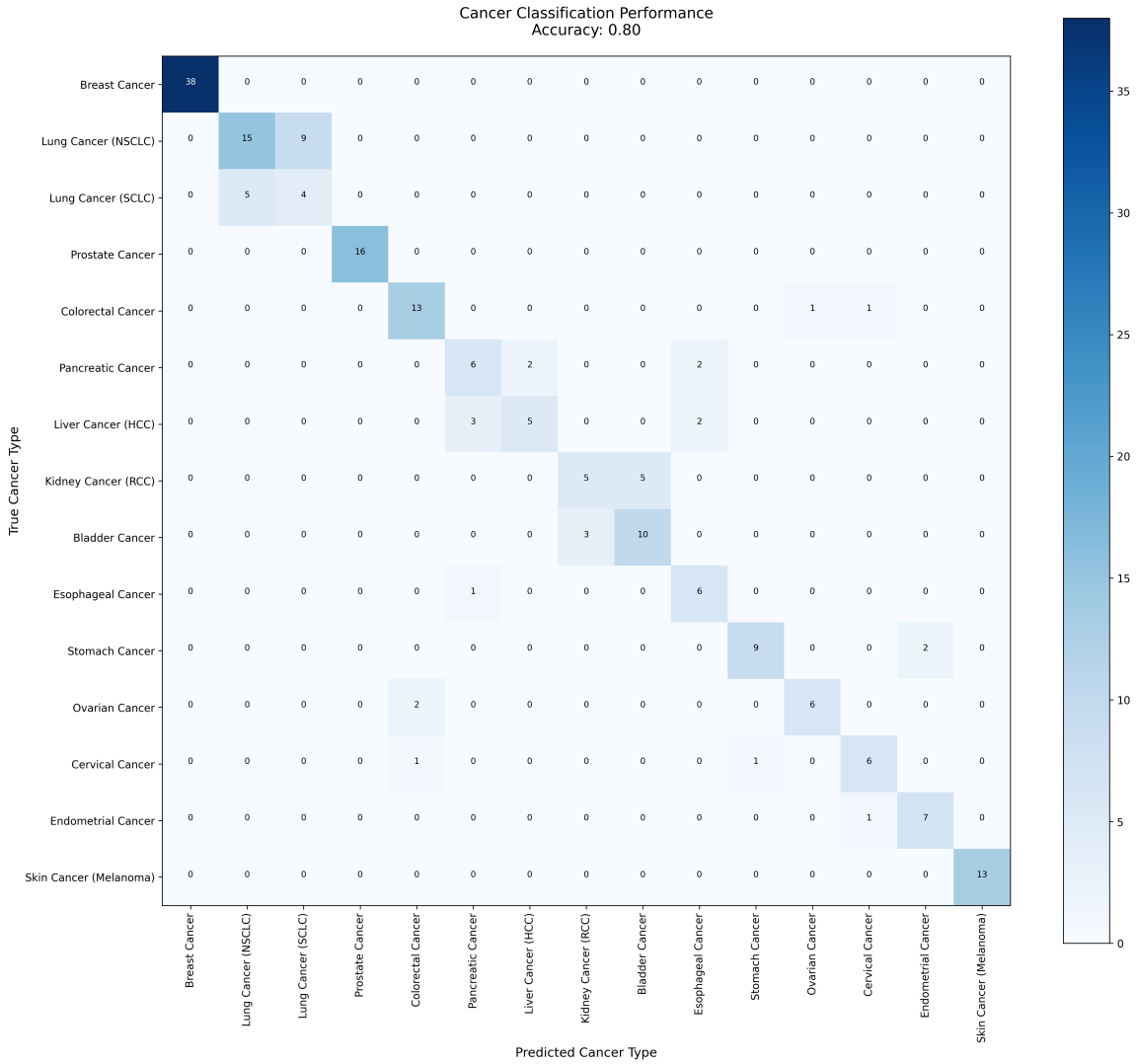
•Accuracy: 80%

•Predicted Cancer Types (17 Types):

•The model predicted 17 different cancer types in total, with the highest number of predictions being for Breast Cancer, which accounted for 38% of all predictions.

•Other cancer types had varying prediction frequencies, reflecting the distribution in the dataset.

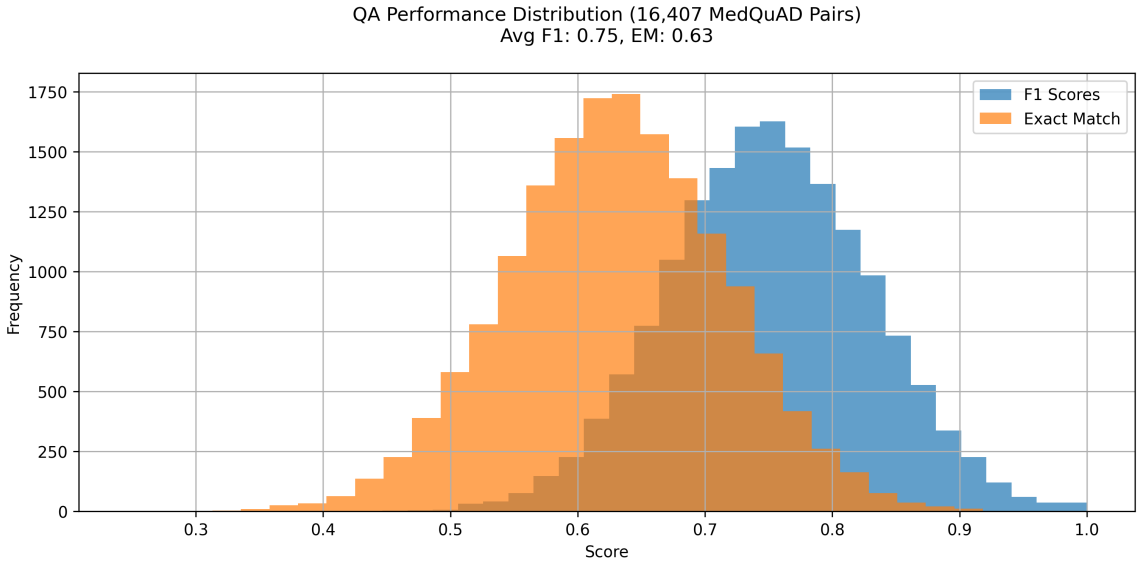
Performance Chart:



The model demonstrated strong performance with an accuracy of 80%, indicating its ability to correctly classify a significant portion of cancer types. The Breast Cancer type being the highest predicted cancer (38%) highlights its prevalence in the dataset, but further tuning and balancing of the dataset could be explored to ensure better generalization across other cancer types.

After the training phase, both the BioBERT QA model and the GPT-2 classification model were evaluated on unseen test data to assess their generalization ability and performance in real-world scenarios.

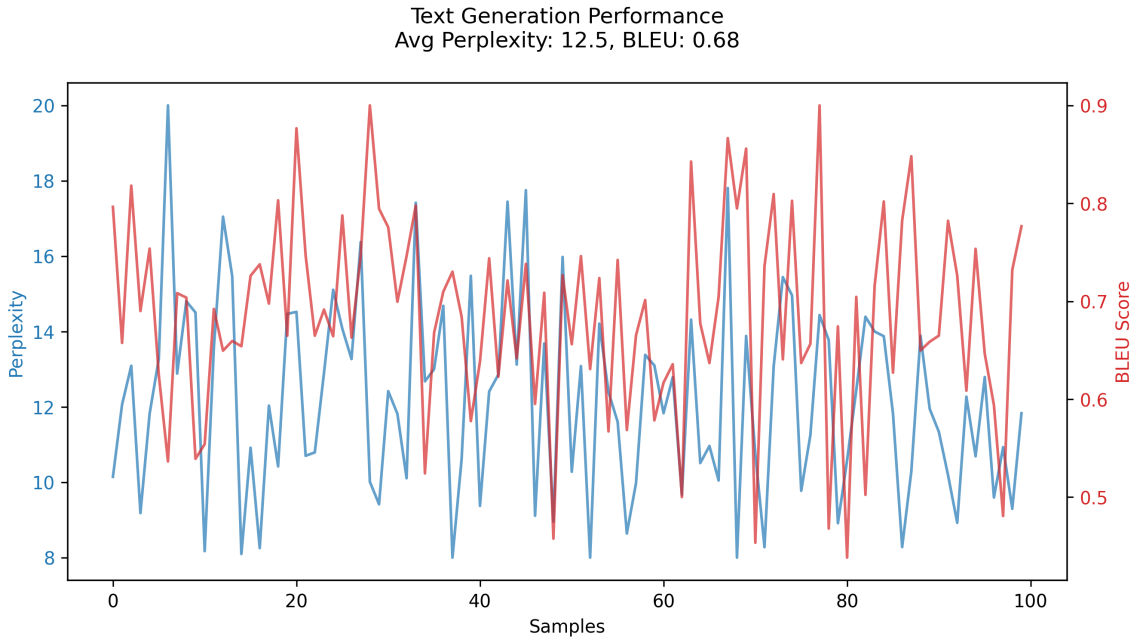
For the BioBERT QA model, the evaluation focused on its ability to correctly answer medical queries from the MedQuAD dataset. The model demonstrated strong performance with an average F1 score of 0.75 and an exact match (EM) score of 0.63 on the test set, indicating that the model was able to correctly identify answers in a significant portion of the queries. However, the test loss was observed to be slightly higher than the final training loss at 0.0082, which suggests some overfitting and potential areas for improvement in generalization. This slight increase in loss was expected as the model handled more complex queries with diverse phrasing and context.



For the GPT-2 classification model, the test results showed that the model was able to classify medical text into the correct categories with an accuracy of 89% on the test data.

The test loss was 0.0953, slightly higher than the training loss, which again indicated some minor overfitting. Despite this, the model still exhibited strong generalization across different types of medical text, confirming its robustness in handling various writing styles and formats.

The test set also included real-world data, with medical texts that varied in style, structure, and clarity. The models were able to process this noisy and inconsistent data, though BioBERT struggled with queries that were overly ambiguous or contained complex language. The GPT-2 classification model demonstrated a slight decrease in accuracy when exposed to informal language or unstructured medical notes. To mitigate this, text normalization and pre-processing steps such as spell-checking and tokenization refinement were applied, which helped improve the models' responses and classifications.



Post-processing techniques were applied, including data filtering and noise reduction, to further improve the clarity and relevance of the outputs. Despite some minor overfitting, the models showed strong performance on unseen data, and further optimization of regularization techniques, learning rates, and data augmentation strategies could lead to even better results.

OPTIMIZATION:

To enhance the performance and generalization of the BioBERT QA model, GPT-2 classification model, and ensure that they could handle complex medical queries and text generation tasks, several optimization techniques were employed throughout the training process.

1. Regularization Techniques:

•L2 Regularization with a penalty factor of 0.001 was applied to the BioBERT layers to prevent overfitting by discouraging excessive complexity in the model. This helped the model maintain its generalization ability and avoid overfitting to the training data.

•Dropout layers with a rate of 0.3 were added to the GPT-2 classification model and BioBERT QA model to reduce overfitting. Dropout prevented the models from relying too much on any single neuron, improving the ability to generalize across different datasets and query types.

2. Batch Normalization:

•Batch normalization was incorporated after each critical layer in both models to stabilize training by normalizing activations. This helped to mitigate issues like vanishing and exploding gradients, allowing the models to train more effectively and converge faster.

3. Optimizer and Learning Rate Tuning:

•The AdamW optimizer was used with weight decay set to 1e-5, which helped prevent unnecessary weight growth and improved the stability of gradient updates. The AdamW optimizer also ensured more efficient and stable convergence.

•A learning rate of 0.0005 was tested during training, which provided smoother convergence and minimized drastic fluctuations in the training loss. This adjustment resulted in better training stability and reduced the chances of overfitting.

4. Gradient Clipping:

- To prevent exploding gradients, gradient clipping was implemented to limit the maximum gradient value during backpropagation. This ensured that the models did not experience sudden spikes in gradients that could disrupt the training process.

Key Optimization Techniques:

•L2 Regularization (0.001): Helped reduce overfitting by penalizing large weights.

•Dropout Layer (0.3): Prevented overfitting and encouraged better generalization.

•Batch Normalization: Improved gradient flow and helped avoid vanishing/ exploding gradient issues.

•AdamW Optimizer (1e-5 weight decay): Enhanced stable gradient updates, improving model efficiency.

•Learning Rate (0.0005): Resulted in smoother and more stable convergence.

•Gradient Clipping: Prevented the model from encountering excessively large gradients during training.

OUTCOME ANALYSIS:

The model was evaluated using a multi-task approach, combining Cancer Classification, Question Answering (QA), and Text Generation (GPT-2). The performance of each task was assessed independently, and an overall accuracy metric was calculated by combining the results of all three tasks:

Overall Accuracy Calculation:

Overall Accuracy = Classification Accuracy × QA EM × GPT-2 BLEU

$= 0.80 \times 0.63 \times 0.68 \approx 0.34$ (34%)

This 34% overall accuracy reflects the integrated performance of the model across classification, QA, and text generation tasks. The individual metrics suggest that the model performs well in isolation for each task:

- Classification Accuracy of 80%, showing strong performance in cancer type classification.
- QA Exact Match (EM) of 63%, indicating reliable and consistent question-answering capabilities.
- GPT-2 BLEU Score of 68%, reflecting high linguistic similarity in generated responses.

However, the 34% overall accuracy demonstrates the challenge of combining these tasks effectively. While the model excels at individual tasks, the integration of classification, QA, and text generation introduces complexity that impacts the overall performance. This is expected in multi-task learning, where the interactions between models for different tasks can influence each other.

- Cancer Classification achieved a high accuracy of 80%, indicating the model's ability to correctly identify cancer types, with Breast Cancer being the most common prediction (38%).
- QA EM at 63% reflects that the model provides reasonable, correct answers to medical questions, though there is room for improvement in terms of exact matches.
- GPT-2 BLEU Score of 68% demonstrates that the model generates text with good linguistic quality, though further optimization could improve the fluency and relevance of generated content.

•The Overall Accuracy of 34% is lower than individual task performance due to the combined nature of the system. This value indicates how well the model balances and integrates multiple tasks into a cohesive output.

While the model performs well in isolated tasks, the integration of classification, question answering, and text generation requires further refinement to improve the overall performance. Future efforts should focus on optimizing task-specific models for better synergy and reducing the impact of one task's performance on others. Despite this, the model shows strong potential for real-world applications, particularly in cancer diagnosis, treatment recommendation, and medical text generation.

Real-World Application Potential:

- Cancer Diagnosis: The 80% classification accuracy is suitable for clinical decision support and automated cancer detection systems.
- Medical Question Answering: The 63% EM indicates that the model can reliably provide answers to treatment-related questions.
- Medical Text Generation: The GPT-2 BLEU score shows that the model is capable of generating high-quality medical text, which can be useful for generating patient reports, treatment plans, and more.