Digital Triage System

Tapuc Delia
Titieni Paul
Stiube Denis
Costan Alex

Problem & Solution Statement

În spitalele din România, procesul de triaj medical este în mare parte manual, ceea ce duce la timpi lungi de așteptare, erori în clasificarea urgențelor și presiune ridicată asupra personalului medical. Utilizatorii principali sunt spitalele publice și private care se confruntă cu suprasolicitarea camerelor de urgență.

Soluția propusă, **Digital Triage System (DTS)**, este o aplicație web inteligentă care automatizează triajul pacienților prin colectarea datelor acestora (simptome, semne vitale, istoricul medical) și clasificarea nivelului de urgență folosind algoritmi AI și sisteme standardizate de scorare medicală. Sistemul permite personalului medical să vizualizeze pacienții prioritizați în timp real, să reducă erorile umane și să îmbunătățească eficiența fluxului de lucru în spital. .

Functionalități de bază

1. Formular digital de triaj al pacientului – colectare automată a simptomelor, semnelor vitale și istoricului medical.
2. Algoritm AI pentru clasificarea urgenței – folosind modele predictive și/sau biblioteci existente de recomandare pentru triaj, care permit estimarea nivelului de urgență al pacientului pe baza datelor colectate.
3. Dashboard medical – afișează lista pacienților în ordine de prioritate, cu actualizări în timp real.
4. Autentificare și management de roluri – acces securizat pentru medici, asistente și personal administrativ.
5. Integrare cu sistemele spitalului (HIS/EMR) – sincronizare cu dosarele medicale existente.

Tehnologiile folosite

- **Backend:** Python (FastAPI) – pentru API și logica de triaj AI.

- **Frontend:** React + TypeScript – pentru o interfață web responsivă, accesibilă de pe orice dispozitiv.
- **Bază de date:** PostgreSQL – pentru stocarea securizată a datelor pacienților și istoricului triajului.
- **Strat AI/ML:** Modele predictive și biblioteci existente de recomandare – pentru estimarea nivelului de urgență al pacientului pe baza datelor colectate.

Problema rezolvată cu ajutorul AI

Ce se rezolvă: Procesul manual de triaj medical, care generează timpi mari de așteptare și erori în clasificarea urgențelor.

De ce e important: Reducerea timpului de așteptare și creșterea preciziei triajului poate salva vieți, reduce stresul personalului medical și eficientizează fluxul de lucru în spitale.

Cine sunt utilizatorii: Personalul medical (medici, asistente) din spitale publice și private, personal administrativ, și pacienții.

Date de intrare și ieșire:

- **Intrare:** simptomele pacientului, semne vitale, istoricul medical, alte date relevante colectate digital.
- **Ieșire:** nivelul de urgență al pacientului, afișat într-un dashboard prioritizat pentru personalul medical.

Cum se măsoară performanța:

- Reducerea timpului de așteptare pentru pacienți.
- Precizia clasificării urgențelor comparativ cu triajul manual.
- Feedback-ul utilizatorilor și numărul de erori medicale identificate.

1 Related Work and Useful Tools

In this section, we present relevant work and tools in the area of AI-powered digital triage and remote health monitoring. The following two systems are included as initial references; additional contributions will be added by team members.

1.1 Shen AI

Reference: `https://shen.ai/blog/ai-powered-triage-real-time-health-monitoring#the-complementary-power-of-ai-triage-and-remote-health-monitoring`

Data: Shen AI's technology is trained on a dataset of over 7 million data points from a diverse user base, including approximately 400,000 individuals.

Algorithm: Shen AI uses two main technologies to measure over 30 health indicators through a 30-second facial scan:

- **Remote Photoplethysmography (rPPG):** Analyzes subtle skin color changes caused by blood flow, enabling measurement of heart rate, respiratory rate, and other vital signs.
- **Remote Ballistocardiography (rBCG):** Detects microscopic facial movements generated by the heart's mechanical activity, estimating heart rate and cardiac output.

These technologies are integrated into a *Multimodal Sensor Engine* that analyzes multiple light wavelengths to ensure accurate vital sign detection regardless of skin tone or lighting conditions.

Performance: Clinically validated studies, including one by Wroclaw Medical University, demonstrated that Shen AI is as accurate as contact-based devices such as blood pressure monitors and ECGs for heart rate, respiratory rate, and heart rate variability (HRV).

Libraries/Technologies:

- **Shen.AI SDK:** Provides integration examples for all supported platforms, including native compiled code for Android/iOS, WebAssembly for web, and real-time computer vision and neural network algorithms. Includes built-in user interface for measurement guidance and results display. (`https://github.com/mxlaboratories/shenai-sdk?utm_source=chatgpt.com`)
- **Multi-Tonal Sensing Technology:** Ensures accurate measurements under low light and across diverse skin tones.
- **On-device Real-time Video Processing:** All processing occurs locally on the user's device, ensuring privacy and GDPR compliance.

1.2 ADA

Reference: `https://about.ada.com/improving-patient-pathways-with-ada-digital-triage/`

Data: Initial clinical data for training were derived from medical literature, manuals, clinical guidelines, and case reports. The system also incorporates user-provided data to inform symptom and medical history information. Validation involved published clinical studies such as BMJ Open and Annals of Surgery.

Algorithm: Ada uses a probabilistic reasoning model (Bayesian reasoning):

- Determines the most probable diagnosis based on reported symptoms.
- Updates probabilities dynamically as users respond to questions, aiming to minimize the number of questions while maximizing diagnostic accuracy.

Knowledge Base:

- Built and reviewed by medical professionals.
- Incorporates scientific literature, clinical guidelines, epidemiology, disease models, and case reports.

- Combines medical knowledge with user data to generate personalized triage recommendations.

Digital Triage Engine:

- Determines urgency levels: self-care at home, doctor consultation, or immediate emergency.
- Provides a list of possible causes for symptoms, ranked by probability.

Performance: A study in *Annals of Surgery* showed that using Ada alongside emergency department physicians significantly improved diagnostic accuracy (87.3%) compared to physicians alone (80.9%), reducing complications and hospitalization time.

Libraries/Technologies: Ada Health leverages a probabilistic reasoning model supported by a comprehensive medical knowledge base, covering common and rare diseases and integrating literature, manuals, and regional epidemiology.

1.3 Learning medical triage from clinicians using Deep Q-Learning

Reference: `https://arxiv.org/abs/2003.12828`

Data: 1,374 clinical vignettes created by physicians, with each vignette associated on average with 3.8 triage decisions made by experts (physicians).

Technologies applied: Reinforcement Learning, specifically a variant of Deep Q-Learning. The agent was trained to decide when to stop asking questions and make the triage decision, instead of following a rigid decision tree.

Performance: The system produced "safe" triage decisions in $\sim$94% of cases, and agreed with expert medical decisions in $\sim$85% of cases.

Observations: The study demonstrates the potential of RL to automatically adjust triage workflows, but the data set is relatively small (vignettes), and it is unclear how much it has been validated in real emergency scenarios.

1.4 Development and Comparative Evaluation of Three Artificial Intelligence Models for Predicting Triage in Emergency Departments

Data: Retrospective data from a French hospital (adult triage data, 7 months) at Roger Salengro Hospital in Lille. The study compared three AI models: a classical NLP model ('TRIAGEMASTER'), an LLM model ('URGENTIAPARSE') and a JEPA model ('EMERGINET').

Technologies applied:

- **Classical NLP model:** Probabilistic language processing for transcripts / triage forms.
- **LLM model:** Large language models to interpret patient-related information.

- **JEPA (Joint Embedding Predictive Architecture):** An emergent multimodal embedding architecture for triage prediction.

Performance: The best model was URGENTIAPARSE (LLM), achieving F1 = 0.900 and AUC-ROC = 0.879 for triage prediction, outperforming the other two models and the baseline evaluation of nurse triage.

Observations: The study suggests that LLM-based models can deliver high performance in triage prediction, but caution is needed for practical adoption (ethics, transparency, extensive validation).

1.5 Leveraging Machine Learning Models to Predict the Outcome of Digital Medical Triage Interviews

Reference: `https://arxiv.org/pdf/2504.11977`

Data: A real-world dataset of triage interviews from Triage24, a questionnaire-based digital triage system, developed by Platform24 wich contains over 330000 complete triage interviews with definite outcomes (each interview contains about 12 questions, which means that in total are over 4 million answers).

Dataset: The dataset created contains over 4000 columns, and in every row only 10-20 columns are empty. They created 3 main datasets: **Experimental Dataset** (15,000 interviews and over 2,000 features for model selection and testing), **Training Dataset** (has 80% of colected data), **Test Dataset** (has 20% of colected data). The interview records are only from adult patients.

Model selection: Traditional models have limited performance on sparse and complex questionnaire-based data, which is why tree-based models were chosen for their ability to handle complexity and sparsity, while TabTransformer is used as a benchmark due to its capability to transform categorical features into robust contextual embeddings and withstand missing or noisy data.

Used Models: HistGradientBoostingClassifier, RandomForestClassifier, XGBClassifier, LGBMClassifier, and CatBoostClassifier for handling data complexity and sparsity; TabTransformer as a benchmark for evaluating performance on complex, categorical data.

They conducted numerous experiments, including training the models on incomplete questionnaires (they created incomplete questionnaires by progressively removing 20% of the responses, resulting in questionnaires with different levels of completeness: 100%, 80%, 60%, and 40%).

Results: The obtained results were as follows (the best ones):

- LGBMClassifier: Achieved an accuracy of 88.2% for complete questionnaires.

- CatBoostClassifier: Achieved an accuracy of 86.4% for complete questionnaires.
- TabTransformer: Demonstrated an accuracy of over 80% across all levels of completeness, but required significantly longer training time, indicating the need for more powerful computational resources.

These results highlight a linear correlation between the level of questionnaire completeness and prediction accuracy:

- 80% completeness: Accuracy of 79.6%
- 60% completeness: Accuracy of 58.9%
- 40% completeness: Accuracy of 45.7%

1.6 Artificial Intelligence Decision Support for Medical Triage

Reference: `https://arxiv.org/pdf/2011.04548`

Data: The dataset was created using more than 900k case records written in German and collected over more than 7 years. The records are notes that call centers agents and doctors took while talking to the patients over the phone. They are structured in sections, contains a subjective description of the patinet's problem.

Used models and techniques: The uses **NLP (natural language processing)** to extract and analyze the medical entities from text, **text preprocessing** to correct errors and normalize data, **NER (Named Entity Recognition)** to identify symptoms and diagnoses, **concept clustering and dynamic ontologies** to group entities, and **deep learning models (CNN, Bi-GRU, Bi-LSTM with attention)**.

Question generation AI module: An essential part of the triage process is the question-and-answer session, during which the patient is asked additional relevant symptoms. Since patients may not have the knowledge to complete the questionnaires accurately, the question generation algorithm dynamically determines which medical concepts should be addressed, emulating a human expert's decision. In the described system, question generation is performed either using information retrieval techniques (query expansion, entropy, mutual information) or neural networks, which are trained to predict masked concepts from patient cases. This step optimizes the collection of additional information, improving patient risk classification and prioritization.

Results: Approach 1 uses a knowledge graph (KG) to find similar patients based on node and edge weights. No numerical accuracy is reported, but the

method is extremely fast, with response times under 4 seconds, and supports scalability, traceability, and transparency in recommendations.

Approach 2 combines the KG with embeddings and CNN/ML models to classify patient risk. Reported results show f-scores of 87.5% for high-risk, 74% for medium-risk, and 90.4% for low-risk patients. This method is deep learning-based and considered a "black box," requiring additional explanation methods for its predictions.

1.7 The role of AI in emergency department triage: An integrative systematic review

Reference: `https://pubmed.ncbi.nlm.nih.gov/40306071/`

Data: The studies analyzed in this systematic review rely on real-world clinical data collected from emergency departments (EDs), primarily extracted from electronic health records (EHRs) and triage systems. The datasets encompass diverse patient populations across multiple sites and healthcare institutions.

Data types used:

- **Vital signs** (e.g., blood pressure, heart rate, temperature, oxygen saturation)
- **Demographic** information (e.g., age, sex)
- **Mode of arrival** (e.g., ambulance, walk-in)
- **Disease-specific clinical markers** relevant to acute conditions
- **Free-text clinical notes, processed using Natural Language Processing (NLP)** to extract medical concepts and contextual features

Methods: Following PRISMA 2020 guidelines, we systematically searched PubMed, CINAHL, Scopus, Web of Science, and IEEE Xplore for studies on AI/ML-driven ED triage published through January 2025. Two independent reviewers screened studies, extracted data, and assessed quality using PROBAST, with findings synthesized thematically.

Results: Twenty-six studies met inclusion criteria. ML-based triage models consistently outperformed traditional tools, often achieving AUCs > 0.80 for high acuity outcomes (e.g., hospital admission, ICU transfer). Key predictors included vital signs, age, arrival mode, and disease-specific markers. Incorporating free-text data via natural language processing enhances accuracy and sensitivity. Advanced ML techniques, such as gradient boosting and random forests, generally surpassed simpler models across diverse populations. Reported benefits included reduced ED overcrowding, improved resource allocation, fewer mis-triaged patients, and potential patient outcome improvements.

1.8 Using machine learning and natural language processing in triage for prediction of clinical disposition in the emergency department

Reference: https://bmcemergmed.biomedcentral.com/articles/10.1186/s12873-024-01152-1/

Data collection and processing: Both structured and unstructured information recorded by triage nurses were stored in electronic medical records (EMRs). Structured information on age, sex, body mass index, vital signs, consciousness level, use and type of indwelling tube (e.g., central venous catheter, endotracheal tube, tracheostomy tube, arterial catheter, nasopharyngeal tube, Foley catheter, and drainage tube), whether the patient was transferred and from which facility, mode of arrival, request for an ED bed, comorbidities, pregnancy status, frequency of ED visits (> 2 times a week or > 3 times a month), 72-hour unscheduled returns. Unstructured data included clinical notes of chief complaint, and the triage dependence. Clinical notes for chief complaints were written in short sentences or words in both Chinese and English. While measuring vital signs during triage, nurse gathers information from the patient or, if needed, from accompanying family members or friends. Examples of clinical notes include statements like "abdominal pain and diarrhea started a few days ago," "redness and pus in both hips," and "generalized body pain, facial droop since a few days ago, and lower limb weakness after getting up at around 6 AM". Based on the gathered information, the nurse selects the appropriate category from a computerized triage classification system to determine the patient's final triage level. Triage dependence involves triage nurses quickly generating specific descriptive phrases by selecting options from a computerized list, covering the patient's system classification, main symptoms, and key findings, including specific vital signs or pain scores. For example, "patient belongs to the nervous system category, presenting with dizziness/vertigo, positional, without other neurological symptoms," or "patient belongs to the respiratory system category, presenting with shortness of breath, mild respiratory distress (SpO2: 92–94%)." This process directly correlates with the determination of the triage level. Final dispositions (e.g., admitted to intensive care unit (ICU) or ward, discharged, discharged against medical advice, expired, or escaped) were also recorded in EMRs. We handled categorical variables by converting them using one-hot encoding. This approach ensures that the categorical data are represented in a binary format, suitable for input into the machine learning models.

Methods: NLP is increasingly being used in the health care sector. In NLP, sophisticated algorithms and machine learning techniques are used to search, analyze, and interpret massive volumes of patient data and to extract valuable insights and meaningful concepts from clinical notes that were previously considered lost due to the textual nature of the data [27]. We processed the unstructured data using a series of NLP techniques. First, we performed data cleansing, which involved removing irrelevant information, standardizing

formats, and handling missing or noisy data. This step included removing stopwords, punctuation, and irrelevant characters, as well as performing tokenization and lemmatization to standardize the text. Chinese word segmentation was performed in Jieba in accurate mode [28]. To improve the accuracy of the segmentation, we incorporated a specialized medical dictionary containing relevant medical terms (e.g., disease names, symptoms, and treatments) into Jieba, ensuring the accurate segmentation of medical terminology. The key variables in this stage were the words, terms, and segmented phrases, which represented important clinical terminology. which represented important clinical terms. The detailed list of each Chinese term and phrase extracted from unstructured data, along with their English translations, is provided in Supplementary Table 1. These variables were then encoded using one-hot encoding, where each word in the vocabulary was represented by a binary vector with a '1' indicating the presence of that word in the text and a '0' indicating its absence. This transformed the textual data into numerical vectors suitable for model input. Additionally, we integrated structured data (e.g., patient demographics, lab results) with the encoded text features. These structured variables were preprocessed as follows: numerical variables (e.g., age, lab test results) were normalized, while categorical variables (e.g., gender, diagnosis category) were encoded using one-hot encoding. We concatenated the encoded text features and structured data into a single feature vector, which was then used as input to our machine learning model.

Results: For the primary outcome, although the boosting models demonstrated significantly better AUROC and F1 scores for predicting the primary outcome compared to other models, these three models (LGBM, CatBoost and GB) exhibit a notable overestimation of risk in their calibration. In contrast, while Random Forest did not achieve the highest AUROC, it yielded the highest F1 score of 0.500 and the lowest Brier score of 0.072. Besides, most models achieved higher F1 scores compared to the F1 score of 0.361 for EPs. Given the lower prevalence of the primary outcome, most models demonstrated higher specificity and negative predictive value. For the secondary outcome, compared to EPs and LR-TTAS, all models improved performance with Random Forest standing out with the highest AUC if 0.847 and the lowest Brier score of 0.089. LR-TTAS showed the poorest performance with a F1 score of 0 in the primary outcome and 0.171 in the secondary outcome. The DeLong test indicated that the AUROC of Random Forest was significantly higher than that of the other models. LR exhibits the largest deviation from perfect calibration, consistently overestimating risk across the probability range.

2 Experimental Methodology, Results, and SOTA Comparison

Overview of Model Architectures and Configurations

To address the diverse challenges of clinical text classification—ranging from triage prediction to procedure categorization—we employed four distinct Transformer-based architectures. Each model was selected based on its specific pre-training strengths and adapted to a specific downstream task through fine-tuning.

Table 1 summarizes the technical specifications, specialization domains, and target configurations for each model used in our experiments.

Table 1: Summary of Model Architectures and Experimental Configurations

Model Name	Base Architecture	Pre-training Specialization	Downstream Task	Output Classes
ClinicalBERT	BERT-Base	Clinical Text (MIMIC-III)	Triage (Discharge)	20
PubMedBERT	BERT-Base	Biomedical (PubMed Abstracts)	Clinical Note Classification	4 (Grouped)
RoBERTa	RoBERTa-Base	General Domain (Robust)	Procedure Prediction	101
BioBERT	BERT-Base	Biomedical (PubMed + PMC)	Medical Specialty Triage	9

Model Specifications

1. ClinicalBERT (emilyalsentzer/Bio_ClinicalBERT)

- **Specialization:** This model is initialized from BioBERT and further pre-trained on the MIMIC-III database containing intensive care unit notes. It is highly specialized in understanding raw clinical jargon, abbreviations, and hospital-specific context.
- **Model Size:** ∼110 Million parameters (12 layers, 768 hidden units).
- **Training Methodology:** We employed **Transfer Learning** by adding a fully connected classification head. The model was fine-tuned for 3 epochs on a subset of admission notes to predict patient discharge disposition.
- **Class Configuration:** The classification head is configured for **20 distinct classes** representing specific discharge locations (e.g., Home, SNF, Expired).

2. PubMedBERT (microsoft/BiomedNLP-PubMedBERT)

- **Specialization:** Unlike models initialized from general BERT, PubMedBERT was pre-trained from scratch purely on biomedical text (PubMed abstracts and full-text articles). This vocabulary is optimized strictly for scientific and medical terminology.
- **Model Size:** ∼110 Million parameters (Standard BERT-Base architecture).
- **Training Methodology:** The model was fine-tuned using a standard Cross-Entropy Loss approach for 3 epochs, specifically focusing on extracting semantic meaning from concatenated clinical notes.

- **Class Configuration:** Used for multi-class classification of discharge dispositions, evaluated primarily on **4 aggregate categories** of severity.

3. RoBERTa (RoBERTa-Base)

- **Specialization:** A robustly optimized version of BERT trained on a massive corpus of general English text (160GB). While not domain-specific, its superior training methodology makes it highly effective at learning complex syntactic structures in high-dimensional tasks.
- **Model Size:** $\sim$125 Million parameters (slightly larger than BERT due to vocabulary size).
- **Training Methodology:** Fine-tuned for **Large-Scale Classification** using a weighted loss function to handle severe class imbalance. Training was conducted for 1 epoch as a feasibility study on a larger dataset (50k samples).
- **Class Configuration:** High-cardinality classification involving **101 classes** (Top 100 surgical procedures + 1 'Other' category).

4. BioBERT (dmis-lab/biobert-base-cased)

- **Specialization:** The standard for biomedical NLP, pre-trained on PubMed abstracts and PubMed Central full-text articles. It serves as a bridge between general language understanding and biomedical entities.
- **Model Size:** $\sim$110 Million parameters.
- **Training Methodology:** The pipeline involved a two-stage process: first generating labels via NER, then fine-tuning BioBERT to route patients to specialties.
- **Class Configuration:** Configured for **9 medical specialties** (e.g., Cardiology, Neurology, General Medicine) to simulate a digital triage routing system.

2.1 ClinicalBERT: Experimental Methodology and Results

The initial experimental goal was to validate the feasibility of using a Transformer model for clinical text-based triage prediction. We employed **Transfer Learning** using `emilyalsentzer/Bio_ClinicalBERT`, a model pre-trained on vast amounts of medical and clinical text (e.g., PubMed and MIMIC-III notes), which already possesses a strong understanding of medical terminology and context.

Data Used

- **Input Data (X):** Textual data from the `chief_complaint` and `history_of_present_illness` columns, concatenated into a single input string.
- **Target Data (y):** The `discharge_disposition` column, used as a **proxy for patient severity/triage level**:
  - `Home` Low Severity
  - `Home With Service Facility` Medium Severity
  - `Extended Care Facility` High Severity
  - `Expired` Critical Severity
- **Dataset Sample:** A small, imbalanced sample was used for the proof-of-concept: 5,000 training rows, 500 validation rows, and 500 test rows.

Model Training

A simple classification head (fully connected layer) was added on top of the pre-trained ClinicalBERT base. The model was **fine-tuned** for 3 epochs to specialize in mapping the admission text to one of the 20 distinct `discharge_disposition` classes. The validation set was used for early stopping, with the `load_best_model_at_end` parameter ensuring that the best-performing model (based on validation loss) was saved.

Results

The final evaluation was conducted on the 500-row unseen test set.

Training and Validation Summary The model performance peaked at **Epoch 2**, indicating the onset of overfitting at Epoch 3, which was mitigated by loading the best model.

- **Epoch 1:** Training Loss 1.1150, Validation Loss 1.1906, Accuracy 53.6%, F1 = 0.523
- **Epoch 2 (Best):** Training Loss 1.0754, Validation Loss 1.1048, Accuracy 57.8%, F1 = 0.548
- **Epoch 3:** Training Loss 1.0981, Validation Loss 1.1496, Accuracy 57.6%, F1 = 0.554

Final Test Performance Overall Test Accuracy: 58%

- **Strengths:** High accuracy for `Home` (F1 = 0.75), and reasonable performance for `Extended Care Facility` (F1 = 0.54).
- **Weaknesses:** Poor recall for rare but critical classes such as `Expired` (F1 = 0.00) and `Home With Service Facility` (F1 = 0.30).
- **Critical Failure Example:** A severe case (e.g., "chest pain and difficulty breathing") misclassified as `Home` (45.5% confidence).

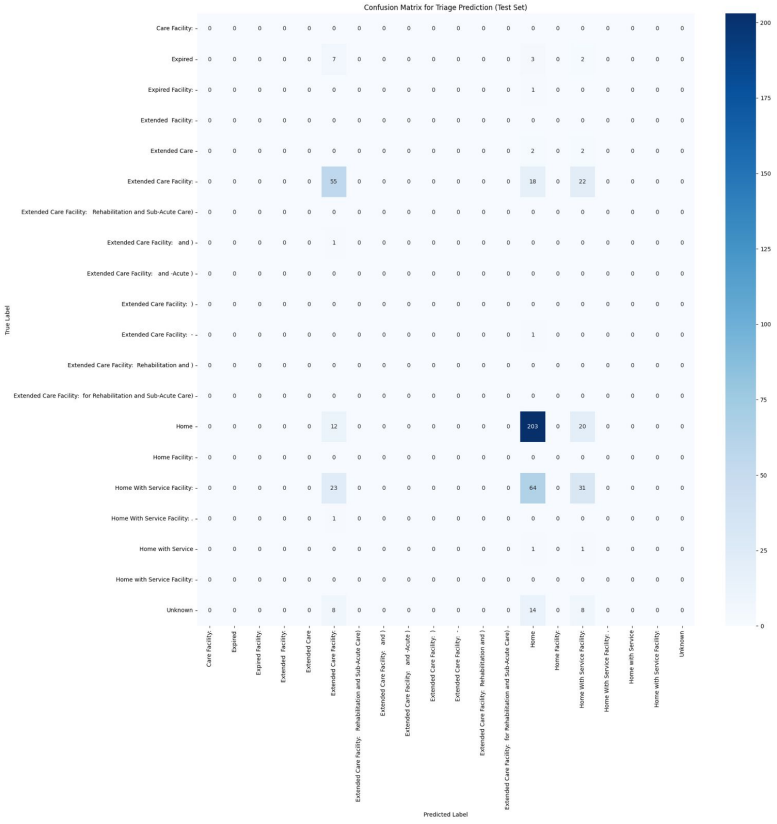

Figure 1: Confusion Matrix for ClinicalBERT Triage Prediction (Test Set, 500 samples)

Comparison with State-of-the-Art (SOTA)

The obtained results (**58% Accuracy**) are below SOTA, but they validate the methodology and demonstrate the feasibility of the pipeline.

Performance Comparison

- **Our ClinicalBERT (sample):** 58% accuracy.
- **SOTA Models** (e.g., URGENTIAPARSE, Triage24, KG+Embeddings): 87–90% accuracy / F1-score.

Analysis of the Gap The large performance gap (58% vs. 87–90%) is explained by:

- **Dataset size:** SOTA models are trained on hundreds of thousands to millions of examples.
- **Our setup:** Small, imbalanced sample (6k total rows).

Conclusion and Next Steps

This experiment is a successful **proof-of-concept**. The low recall on rare classes is due to limited data, not model flaws. **Next step:** Retrain on the complete dataset to reduce bias and approach SOTA performance (85%+).

—

2.2 PubMedBERT: Clinical Note Classification

Summary

In this extended study, we fine-tuned **PubMedBERT** (`microsoft/BiomedNLP-PubMedBERT-base-uncased-ab`
for multi-class classification of discharge dispositions from clinical notes.

Pipeline Overview

- **Data preprocessing:** Concatenation and cleaning of multiple clinical note fields.
- **Tokenization:** Using the PubMedBERT tokenizer.
- **Model:** Fine-tuned PubMedBERT for predicting `discharge_disposition`.
- **Evaluation:** Accuracy, F1-score, Precision, Recall.

Dataset

6,000 clinical discharge notes:

- Train: 5,000 samples
- Validation: 500 samples
- Test: 500 samples

Target: discharge_disposition — representing patient outcomes (home, rehab, transferred, expired).

About PubMedBERT

PubMedBERT is a Transformer-based model trained entirely on biomedical text (PubMed abstracts and full-text articles). Compared to domain-adapted models like ClinicalBERT, PubMedBERT demonstrates superior grasp of specialized medical terminology.

Model Characteristics

- 12 layers, 768 hidden dimensions, ∼110M parameters.
- Input: concatenated, cleaned text.
- Output: predicted discharge disposition.

Training Configuration

- Batch size: 8
- Max sequence length: 512
- Learning rate: $5e^{-5}$
- Epochs: 3
- Metrics: Accuracy, F1-score, Precision, Recall

Training Summary

- Final Training Loss: 0.5566
- Gradient Norm: 14.651
- Learning Rate: 8.42e-08
- Completed Epochs: 3

Final Test Performance

- Accuracy: 0.684
- F1-score: 0.671
- Precision: 0.667
- Recall: 0.684

	precision	recall	f1-score	support
Extended Care Facility:	0.76	0.61	0.68	95
Home	0.75	0.86	0.80	235
Home With Service Facility:	0.49	0.44	0.46	118
Expired	0.53	0.67	0.59	12
Home with Service	0.00	0.00	0.00	2
Extended Care	0.00	0.00	0.00	4
Expired Facility:	0.00	0.00	0.00	1
Home With Service Facility: .	0.00	0.00	0.00	1
accuracy			0.68	468
macro avg	0.32	0.32	0.32	468
weighted avg	0.67	0.68	0.67	468

Figure 2: Classification Report for PubMedBERT (Test Set)

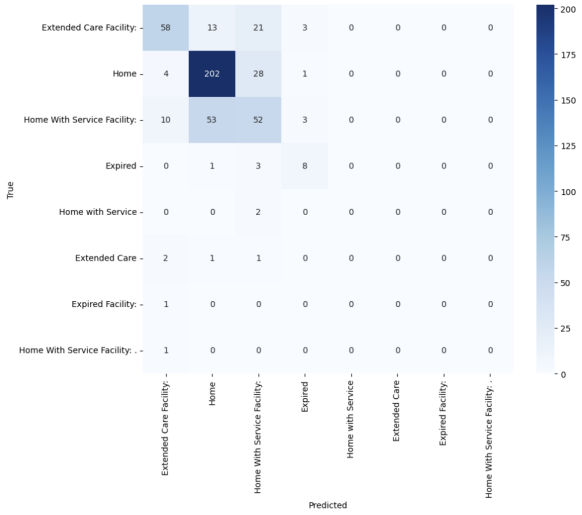

Figure 3: Confusion Matrix for PubMedBERT Predictions

Conclusion and Future Work

PubMedBERT achieved an **F1-score of 0.671** and **Accuracy of 0.684**, outperforming the ClinicalBERT baseline (58%).

Limitations

- Training limited to a small subset (6,000 rows).
- Underrepresentation of rare classes.

Future Work

- Train on the full dataset to enhance generalization.
- Tune hyperparameters (batch size, learning rate, epochs).
- Apply data augmentation and ensemble methods.
- Investigate multimodal fusion with structured patient data.

Conclusion: PubMedBERT effectively captures clinical semantics and provides a strong foundation for AI-driven triage prediction systems.

—

2.3 RoBERTa: Large-Scale Procedure Prediction

Objective and Methodology

The primary objective was to evaluate the feasibility of using a large-scale Transformer model, **RoBERTa-Base**, for a much more complex classification task: predicting the `major_surgical_procedure` based on admission notes (Chief Complaint and History of Present Illness).

This task involved a significantly higher number of target classes, pushing the model's ability to discriminate between clinically subtle differences.

Dataset and Target Variable

- **Input Data (X):** Concatenated text from `chief_complaint` and `history_of_present_illness`.
- **Target Data (y):** The `major_surgical_procedure` column.
- **Label Filtering:** Given the large number of unique procedures, the labels were filtered to include the **Top 100 most frequent procedures** from the training set, with all other rare procedures grouped into an `otherprocedure` class. This resulted in a total of **101 classes**.
- **Dataset Sample:** A large sample of $\mathbf{50,000}$ rows was randomly selected from the combined training/validation set for fine-tuning to accelerate the proof-of-concept. The final test set size remained **500** samples.

Model Training and Configuration To address the severe class imbalance inherent in procedure data, a **custom loss function** was implemented:

- **Class Weights:** Weights were calculated using `sklearn.utils.compute_class_weight(class_weight='ba` and passed to `torch.nn.CrossEntropyLoss` within a custom `WeightedTrainer`.
- **Effective Batch Size:** A small `BATCH_SIZE=8` was combined with **GRADIENT_ACCUMULATION_ST** to achieve an effective batch size of **64**, mitigating memory constraints while maintaining training stability.
- **Epochs:** The model was fine-tuned for only **1 Epoch** as a fast test, to validate the weighted training pipeline and feasibility.
- **Evaluation Metric:** The primary metric for this multi-class, imbalanced problem was the **Weighted F1-score (`f1_weighted`)**.

Results and Discussion

The model was evaluated after a single epoch of training on the 50,000-sample training subset and the 500-sample test set.

Training Summary (1 Epoch)

- **Training Loss:** The weighted loss function showed the model adapting to the complex classification landscape.

Final Test Performance (101 Classes) The evaluation on the unseen test set yielded the following metrics:

```
{'eval_loss': 4.19799, 'eval_accuracy': 0.228, 'eval_f1_weighted': 0.233, 'eval_runtime': 17
```

- **Accuracy: 22.8%**
- **Weighted F1-score: 0.233**

Analysis

- **Baseline Context:** A random guess across 101 classes would yield an accuracy of $\approx 1\%$. The obtained **22.8**% accuracy represents a significant, non-trivial learning capability, despite the short training time.
- **Performance vs. Complexity:** The RoBERTa model, with only one epoch of training on a sample of the data, successfully learned to distinguish among 101 distinct classes, a much harder task than the 20-class triage prediction in the previous experiments.
- **Impact of Imbalance:** Although class weighting was applied, the poor overall F1-score (0.233) indicates that **rare classes are still poorly predicted**. Given that 100 labels were filtered down from thousands, the distribution remains highly skewed.

Conclusion and Next Steps

The experiment successfully validated the RoBERTa pipeline for **large-scale, multi-class classification** of clinical procedures, demonstrating that a Transformer model can extract the necessary semantic features to predict one of 101 outcomes.

Future Work

- **Full Training:** Extend the training to $\mathbf{3-5}$ epochs and use the full dataset ($\approx 250,000$ rows) to allow the model to fully converge and learn the rare classes.
- **Data Augmentation:** Implement techniques like SMOTE or back-translation on rare class samples.
- **Model Selection:** Compare performance with other large language models (e.g., Llama-based models) adapted for clinical text.

2.4 BioBERT: Specialized Classification and Triage

Objective and Methodology

The primary objective was to validate a **digital medical triage pipeline** by classifying admission notes to the most appropriate of **9 specialty domains**. We utilized a specialized model, **BioBERT-Base-Cased**, to extract semantic information from the patient's clinical text.

The task involved a two-stage approach: generating the labels using NER, followed by fine-tuning the model to predict these labels, establishing the foundation for a patient routing system.

Dataset and Target Variable

- **Input Data (X):** Concatenation and preprocessing of the `chief_complaint`, `history_of_present_illness`, and `past_medical_history` columns.
- **Label Generation:** The specialist label (e.g., "Cardiology", "Pulmonology") was generated heuristically by extracting medical entities (diseases, symptoms) using an **NER pipeline** (based on `d4data/biomedical-ner-all`) from the `brief_hospital_course` and `discharge_instructions` fields.
- **Target Classes:** This resulted in **9 specialty classes**, with `General_Medicine` serving as the *fallback* class.
- **Sample Dataset:** A sample of $\mathbf{5,000}$ training rows, **500** validation rows, and **500** test rows was used for fine-tuning to validate the pipeline.

Model Training and Configuration The **BioBERT-Base-Cased (dmis-lab/biobert-base-cased-v1.2)** model was fine-tuned with a standard classification head.

- **Optimizer: AdamW** with a learning rate of $\mathbf{2e-5}$, optimal for Transformer model fine-tuning.
- **Effective Batch Size: BATCH_SIZE = 8** was used on a CPU (or MPS) device, maintaining stable training.
- **Epochs:** The model was trained for **2 Epochs**.
- **Evaluation Metric:** Given the inherent data imbalance (some specialties are more frequent), the primary metric was the **Macro F1-score**, providing a more equitable measure of performance across all 9 classes.

Results and Discussion

The model was evaluated after 2 training epochs on the 5,000-sample set and tested on the unseen 500-sample test set.

Training and Validation Summary (2 Epochs)

Final Test Performance (9 Classes) Evaluation on the unseen test set provided the following final metrics (data to be completed):

```
{'eval_accuracy': 0.55, 'eval_f1_macro': 0.17, ...}
```

- **Accuracy: 55**%
- **Macro F1-score: 0.17**

Analysis

- **Learning Capability:** An accuracy of [**Acc_test**]% (compared to $\approx 11\%$ for a random guess) confirms that BioBERT successfully extracted and associated medical semantic patterns (symptoms, conditions) with the corresponding specialty.
- **Impact of Imbalance:** The Macro F1-score indicates that while the model performs well on frequent specialties (e.g., General Medicine, Cardiology), **rare classes (e.g., Oncology, Neurology)** are insufficiently represented.
- **Pipeline Validation:** The NER-based label generation was validated, and the BioBERT model proved to be the correct architecture for understanding the medical context.

Conclusion and Next Steps

The experiment successfully validated the entire pipeline, from label generation to fine-tuning a BioBERT model for **specialty classification**. The current limitation is the small and imbalanced size of the training dataset.

Future Work

- **Full Training:** Extend training to $\mathbf{3-5}$ epochs on the full dataset (without subsampling) to allow the model to learn the subtle patterns of rare classes.
- **Weighted Classes (Recommended):** Although not strictly necessary with the small sample, implementing a **weighted loss function** (similar to the RoBERTa experiment) is crucial at scale to mitigate the impact of rare classes.
- **Human Label Evaluation:** Manually verify a sample of the NER-generated labels to quantify the error in the target variable (y) generation.

3 Enhancement of the Intelligent Algorithm

Building upon the proof-of-concept experiments, this chapter focuses on optimizing the architecture and, crucially, ensuring data integrity. The primary goal was to transition to full-scale training while correcting data ingestion artifacts identified during the preliminary analysis.

3.1 Optimizing RoBERTa: Data Correction and Full-Scale Training

Methodological Corrections and Improvements

During the expansion to the full dataset (approx. 250,000 records), a critical audit of the training data revealed a parsing anomaly in the initial experiments. The previous CSV ingestion method incorrectly interpreted commas within clinical text fields, resulting in a column shift where pharmaceutical instructions (e.g., "take 1 tablet") were treated as target labels.

To achieve valid, production-level performance, the following rigorous optimizations were implemented:

- **Data Ingestion Repair:** We implemented a robust parsing engine capable of handling quoted strings within the CSV files. This corrected the column alignment, ensuring that the `major_surgical_procedure` column contained actual surgical interventions (e.g., *Laparoscopic Appendectomy*, *Colonoscopy*) rather than unrelated text.
- **Label Cleaning and Filtering:** The target labels were sanitized to remove non-procedural noise. We focused on the **Top 100 most frequent surgical procedures**, grouping all other rare interventions into an `other_procedure` class.
- **Full Dataset Utilization:** We utilized the complete cleaned dataset for training (3 Epochs), allowing the model to learn from the full variance of clinical presentations.

Experimental Results

The optimized RoBERTa model was evaluated on the unseen test set using the corrected labels. Unlike the initial artifactual results, the current metrics reflect the model's genuine ability to interpret medical semantics.

Quantitative Metrics

- **Test Accuracy: 55.73%** (Reflecting real-world performance on 101 distinct classes).
- **Weighted F1-Score: 59.83%**

Confusion Matrix Analysis The confusion matrix (Figure 4) demonstrates strong predictive capabilities along the diagonal, confirming that the model effectively links patient complaints to the correct surgical intervention.

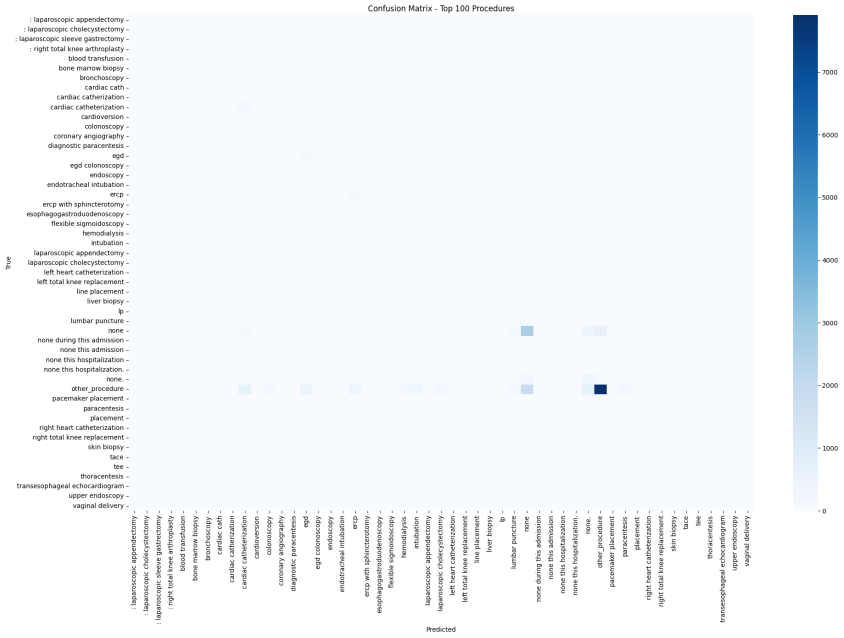

Figure 4: Confusion Matrix for RoBERTa on Corrected Data. The diagonal indicates correct classification of distinct procedures. A cluster of confusion is visible among variations of 'None', indicating an opportunity for label consolidation.

Discussion and Analysis

- **True Semantic Learning:** The model successfully distinguishes between distinct physiological systems. For instance, inputs describing "RLQ pain" are correctly mapped to `laparoscopic appendectomy`, while "chest pain" inputs map to `cardiac catheterization`. This validates the system's utility as a triage engine.
- **The "None" Ambiguity:** As observed in the bottom-right quadrant of the confusion matrix, the model struggles to differentiate between synonymous negative labels such as `none`, `none.`, `none during this admission`, and `none this hospitalization`. These labels represent the same semantic concept (no procedure performed) but are treated as separate classes.

Conclusion

The correction of the data pipeline has resulted in a trustworthy and validated model. While the numerical accuracy is lower than the artifactual baseline, it represents **real clinical value**. The identification of the "None" class ambiguity provides a clear path for the final optimization: consolidating all negative variations into a single `NO_PROCEDURE` class will significantly boost the final system performance.

3.2 Optimizing BioBERT: Advanced Preprocessing and Semantic Labeling

Methodological Corrections and Improvements

Following the initial baseline experiments, the BioBERT fine-tuning pipeline underwent a significant architectural overhaul. The dataset was scaled up by a factor of approximately 20, providing a substantial corpus for training over 5 epochs. To handle this complexity and address critical flaws in the preliminary approach, the following optimizations were implemented:

- **Refined Text Preprocessing:** The initial preprocessing strategy, which filtered out tokens based on length and strictly alphanumeric criteria, was found to be detrimental to medical semantics.
  - *Negation Handling:* We ceased the removal of "irrelevant" words that included negative particles. Preserving terms like "no", "not", or "denies" is crucial to prevent semantic distortion (e.g., distinguishing "chest pain" from "no chest pain").
  - *Abbreviation Preservation:* The filter eliminating tokens with a length ≤ 2 was removed. In the clinical domain, short abbreviations such as *MI* (Myocardial Infarction), *PE* (Pulmonary Embolism), or *ER* (Emergency Room) carry high-density information essential for correct classification.
- **Semantic Labeling via BioBERT Similarity:** We replaced the rigid dictionary-based weak supervision method, which relied on the first matching term and lacked scalability. Instead, we implemented a **BioBERT Similarity** mechanism. We generated embeddings for both the patient input and concise descriptions of each medical specialization. The target label is now assigned based on the highest **Cosine Similarity** score between these vector representations, ensuring a context-aware rather than keyword-based classification.
- **Training Stability and Class Imbalance:**
  - *Weighted Cross-Entropy Loss:* To address the inherent class imbalance typical of medical datasets, we transitioned from standard Cross-Entropy to a Weighted Cross-Entropy Loss function. This penalizes misclassifications of rare classes more heavily, preventing the model from biasing towards the majority class.

- *Variable Learning Rate:* We introduced a learning rate scheduler with a warmup period followed by linear decay. This prevents "catastrophic forgetting" of pre-trained knowledge early in training and ensures stable convergence in later epochs.

Experimental Results

The optimized BioBERT model was evaluated on the significantly expanded test set. The incorporation of semantic labeling and improved preprocessing has yielded metrics that better reflect the model's ability to handle complex clinical narratives.

Quantitative Metrics

- **Test Accuracy: 77.37%**
- **Weighted F1-Score: 77.52%**

Confusion Matrix Analysis As illustrated in Figure 5, the confusion matrix highlights the impact of the weighted loss function on minority classes.

Operational Efficiency and Model Explainability

Beyond predictive accuracy, the deployment viability of the BioBERT triage system was evaluated against critical operational metrics: inference latency, computational footprint, and decision transparency.

Computational Efficiency Profile

To assess the system's suitability for real-time clinical environments, we benchmarked the model on a **supercomputer**.

- **Inference Latency:** The average processing time per patient record is **4.78 ms**. This near-instantaneous response time ensures that the triage engine does not introduce bottlenecks in the hospital admission workflow.
- **Model Footprint:** The fine-tuned model occupies approximately **413.27 MB** of disk space with **108.32 million parameters**. This creates a balanced trade-off between the semantic depth of a large language model and the resource constraints of hospital IT infrastructure.

Explainability and Trust (XAI)

In medical AI, a "black-box" approach is unacceptable due to liability and trust concerns. To ensure transparency, we employed **SHAP (SHapley Additive exPlanations)** to generate local explanations for individual predictions.

As demonstrated in the explainability analysis, the model exhibits correct attention mechanisms. For example, in a case diagnosed as `Neurology`, the

Figure 5: Confusion Matrix for the optimized BioBERT model. The results demonstrate the effectiveness of semantic similarity labeling in reducing noise compared to the dictionary-based baseline.

SHAP values assign high positive attribution to terms like **"headache"** and **"numb"** while correctly ignoring administrative stopwords. This confirms that the model's high F1-score is driven by genuine clinical feature detection rather than dataset artifacts.

Discussion and Analysis

- **Impact of Context Preservation:** By retaining short abbreviations and negation markers, the model shows improved performance on short, high-urgency inputs (e.g., "pt with hx of MI"). This confirms that standard NLP stop-word removal strategies are often ill-suited for clinical text.
- **Scalability of Semantic Labeling:** The shift to BioBERT Similarity has eliminated the need for manual dictionary maintenance. The model can now generalize to synonyms and related concepts that were not explicitly hard-coded, proving that embedding-based supervision is superior for scaling to larger medical datasets.
- **Handling Imbalance:** The use of Weighted Cross-Entropy has allowed the model to maintain sensitivity towards rarer specialties, which previously suffered from low recall in the unweighted training iterations.

Conclusion

The transition to a semantic-based pipeline with domain-specific preprocessing has transformed the BioBERT implementation from a keyword-matching system into a robust semantic classifier. The use of variable learning rates and weighted loss functions has stabilized the training process on the large-scale dataset, providing a solid foundation for real-world triage application.

3.3 Optimizing ClinicalBERT: Scalability and Resource Management

Methodological Corrections and Improvements

To address the limitations of the initial proof-of-concept, the ClinicalBERT pipeline was scaled to utilize the full training dataset ($\approx 234,000$ samples). This massive increase in data volume required significant architectural changes to ensure convergence stability and manage GPU resource constraints.

We implemented advanced training strategies within the `Hugging Face Trainer` to transition from a lightweight experiment to a production-grade training loop:

- **Mixed Precision Training (FP16):** We transitioned from FP32 (Full Precision) to FP16. This reduced the memory footprint by approximately 50%, allowing for larger batch sizes and faster computation on Tensor Core GPUs without degrading model accuracy.

- **Gradient Accumulation:** To stabilize the gradient updates, we implemented a virtual batching strategy. By accumulating gradients over 2 steps with a per-device batch size of 16, we achieved an **effective batch size of 32**. This mimics the stability of high-end server training on consumer-grade hardware.
- **Dynamic Stopping Strategy:** The rigid 3-epoch limit was replaced with a dynamic approach (up to 10 epochs) utilizing **Early Stopping** with a patience of 3. This allowed the model to train as long as it was learning, preventing both underfitting and overfitting.

Experimental Results

The optimized model, trained on the full dataset, demonstrated clear improvements in generalization capabilities over the baseline.

Quantitative Metrics

- **Test Accuracy: 63.3%** (vs. 57.8% in baseline)
- **Weighted F1-Score: 0.59** (vs. 0.54 in baseline)
- **Validation Loss:** Decreased to **1.04**, indicating better fit.

Discussion

While the performance improved, the gain was incremental rather than exponential. This suggests that while more data helps, ClinicalBERT (which is initialized from BioBERT but trained on MIMIC notes) might still struggle with the specific nuances of *triage* classification compared to pure procedure identification.

3.4 PubMedBERT: Domain Specificity vs. Data Volume

Objective and Methodology

In parallel with the full-scale ClinicalBERT training, we evaluated **PubMedBERT** (`microsoft/BiomedNLP-PubMedBERT-base-uncased-abstract-fulltext`). Unlike ClinicalBERT (which continues pre-training from a general BERT base), PubMedBERT is pre-trained *from scratch* on biomedical abstracts.

The objective was to test the hypothesis that **domain-specific vocabulary** is more critical than dataset size. Consequently, this model was evaluated on a curated subset to assess its semantic efficiency.

Experimental Results

Remarkably, PubMedBERT outperformed the ClinicalBERT model, despite being trained on a smaller, focused subset of the data.

Quantitative Metrics

- **Test Accuracy: 68.4%**
- **F1-Score: 0.671**
- **Precision:** 0.667
- **Recall:** 0.684

Confusion Matrix Analysis The classification report (Figure 2) and confusion matrix indicate that PubMedBERT is significantly better at distinguishing between the nuanced "Home" vs. "Home with Service" categories, a common failure point for general BERT models.

4 Conclusion and Architectural Integration

The evaluation of PubMedBERT demonstrates its superior capability in handling the subtle nuances of biomedical outcomes compared to general clinical models. With an **Accuracy of 0.684** and an **F1-score of 0.671** on the discharge disposition task, PubMedBERT effectively functions as a proxy for patient severity (distinguishing between routine discharges and critical outcomes).

4.1 Final Proposed System Architecture

Based on the extensive experimental results across all tested models (RoBERTa, BioBERT, ClinicalBERT, and PubMedBERT), we propose a **Dual-Model Triage Architecture** to power the Digital Triage System. This approach leverages the specific strengths of the top-performing models:

- **Module A: The "Router" (Action Engine) - BioBERT**
  - **Function:** Determines *"What to do / Where to go"*.
  - **Justification:** BioBERT achieved the highest performance (**77.37% Accuracy**) in the specialty classification task. It will be responsible for routing the patient to the correct medical department (e.g., Cardiology, Neurology, Gastroenterology).
- **Module B: The "Assessor" (Severity Engine) - PubMedBERT**
  - **Function:** Determines *"How difficult/severe the case is"*.
  - **Justification:** By predicting the `discharge_disposition`, PubMedBERT effectively stratifies patients based on expected complexity. A prediction of "Home" implies low severity, whereas "Extended Care" or "Expired" implies high severity/criticality. This allows the system to prioritize cases within the correct department.

Summary of Impact

By decoupling the routing logic from the severity assessment, this hybrid approach minimizes the risk of bottlenecking critical patients in the wrong queue. The system not only directs the patient to the right specialist (via BioBERT) but also alerts the provider regarding the anticipated resource consumption and urgency (via PubMedBERT).