Classification for Medical Transcription Specialties

The goal here is to build a model that looks at medical transcriptions and figures out which medical specialty they belong to, like "Surgery" or "Cardiology."

My approach:

I started with understanding the data, cleaned it up, did some exploratory data analysis, built a simple baseline, then used language models also. For the LM part, I did internal (using BioBERT locally for embeddings and a classifier) and external (calling a general model via API for zero-shot help). I compared fine-tuning BioBERT to the baseline, and did EDA on train data and results. I chose BioBERT because it's pre-trained on medical stuff, and XGBoost for internal because it's good at handling imbalanced data.

Dataset Understanding and Preprocessing :

The dataset is a CSV file called mtsamples.csv from Kaggle, with 4,999 rows of medical reports and 6 columns. It's for classifying specialties based on transcriptions.

Labels are multi-class (40 labels)

- Loaded the CSV and checked basics (shape, info, unique specialties, distribution).
- Handled missing values: Dropped rows without transcription, handled null values to keep samples.
- Removed duplicates
- Text cleaning: Lowercased, removed non-letter chars, tokenized, removed stopwords.
- Combined cleaned_transcription and cleaned_keywords into features for better input.
- Split into train/validation: 80/20 with stratification to keep class balance.

Train/Fine-Tune on Domain-Specific Dataset :

I trained a baseline, then fine-tuned BioBERT for domain adaptation.

Baseline (TF-IDF + Logistic Regression):

- Steps: Vectorized text with TF-IDF (5,000 features, bigrams), trained Logistic Regression with balanced weights.
- Results : Accuracy - 0.41, Macro F1 - 0.45

Fine-Tuning BioBERT:

- Tokenized data, loaded BioBERT for classification, used weighted loss for imbalance, trained with Trainer (5 epochs, batch 8, LR 2e-5).
- Results : Accuracy - 0.69, Macro F1 - 0.67

Incorporate Language Model Internally and Externally

- **Internal**: Used BioBERT embeddings (mean pooling for context), resampled with SMOTE, trained XGBoost.
- **External**: Called BART-large via API for zero-shot scores, trained Logistic Regression on them.

Evaluate Effectiveness of Fine-Tuning vs. Pre-Trained Baseline

- **Baseline**: Pre-trained TF-IDF + Logistic (end-to-end, no LM) is fast but shallow - ignores context, accuracy 0.41, F1 0.45.
- **Fine-Tuning**: Adapts BioBERT to domain accuracy 0.69, F1 0.67 - more effective for medical NLP as it learns semantic patterns.
- **Comparison**: Fine-tuning outperforms baseline by accuracy, better on imbalance (via weights). Baseline good for quick tests, fine-tuning for production.