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.

OUTPUTS :

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4999 entries, 0 to 4998
Data columns (total 6 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Unnamed: 0         4999 non-null   int64
 1   description        4999 non-null   object
 2   medical_specialty  4999 non-null   object
 3   sample_name        4999 non-null   object
 4   transcription      4966 non-null   object
 5   keywords           3931 non-null   object
dtypes: int64(1), object(5)
memory usage: 234.5+ KB
None
Unique Medical Specialties: 40
Specialty Distribution:
 medical_specialty
Surgery                            1103
Consult - History and Phy.          516
Cardiovascular / Pulmonary          372
Orthopedic                          355
Radiology                           273
General Medicine                    259
Gastroenterology                    230
Neurology                           223
SOAP / Chart / Progress Notes       166
Obstetrics / Gynecology             160
Urology                             158
Discharge Summary                   108
ENT - Otolaryngology                 98
Neurosurgery                         94
Hematology - Oncology                90
Ophthalmology                        83
Nephrology                           81
Emergency Room Reports               75
Pediatrics - Neonatal                70
Pain Management                      62
```

Handling Missing Values

```
print("\nMissing Values:\n", df.isnull().sum())
df = df.dropna(subset=['transcription'])
df['keywords'].fillna('', inplace=True)
df['description'].fillna('', inplace=True)
```

```
Missing Values:
 Unnamed: 0             0
description            0
medical_specialty      0
sample_name            0
transcription         33
keywords            1068
```

Text Cleaning

```
[8] def clean_text(text):
        text = text.lower()
        text = re.sub(r'[^a-z\s]', '', text)
        tokens = word_tokenize(text)
        stop_words = set(stopwords.words('english'))
        tokens = [word for word in tokens if word not in stop_words]
        cleaned_text = ' '.join(tokens)
        return cleaned_text
```

```
[9] df['cleaned_transcription'] = df['transcription'].apply(clean_text)
    df['cleaned_keywords'] = df['keywords'].apply(clean_text)
    df['cleaned_description'] = df['description'].apply(clean_text)
```

```
[10] print("\nSample Cleaned Transcription:\n", df['cleaned_transcription'].iloc[0])
     df['features'] = df['cleaned_transcription'] + ' ' + df['cleaned_keywords']
```

```
Sample Cleaned Transcription:
 subjective yearold white female presents complaint allergies used allergies lived seattle thinks worse past tried claritin zyrtec w
```

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

BASELINE MODEL OUTPUT :

```
LogisticRegression
LogisticRegression(class_weight='balanced', max_iter=1000,
                   multi_class='multinomial')
```

```
[21] y_pred = clf.predict(X_val_tfidf)
```

```
[22] accuracy = accuracy_score(y_val, y_pred)
     f1 = f1_score(y_val, y_pred, average='macro')
     print(f"Baseline Accuracy: {accuracy:.4f}")
     print(f"Baseline Macro F1: {f1:.4f}")
     print(classification_report(y_val, y_pred))
```

```
Baseline Accuracy: 0.4105
Baseline Macro F1: 0.4511
                                precision    recall  f1-score   support

         Allergy / Immunology       0.17      1.00      0.29         1
                      Autopsy       1.00      1.00      1.00         2
                   Bariatrics       0.40      0.50      0.44         4
    Cardiovascular / Pulmonary       0.49      0.58      0.53        74
                 Chiropractic       0.10      0.33      0.15         3
     Consult - History and Phy.     0.34      0.12      0.17       103
     Cosmetic / Plastic Surgery     0.27      0.80      0.40         5
                    Dentistry       0.44      0.80      0.57         5
```

FINE-TUNED BIOBERT MODEL OUTPUT :

[2485/2485 11:35, Epoch 5/5]

Epoch	Training Loss	Validation Loss	Accuracy	F1
1	No log	3.124861	0.308853	0.162054
2	3.549100	1.600922	0.611670	0.486073
3	2.326000	1.230245	0.687123	0.600291
4	1.313600	1.121577	0.689135	0.653801
5	0.967600	1.106459	0.688129	0.653952

```
TrainOutput(global_step=2485, training_loss=1.7958339883048289, metrics={'train_runtime': 728.4671, 'train_samples_per_second': 27.263, 'train_steps_per_second': 3.411, 'total_flos': 5227168391331840.0, 'train_loss': 1.7958339883048289, 'epoch': 5.0})
```

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.

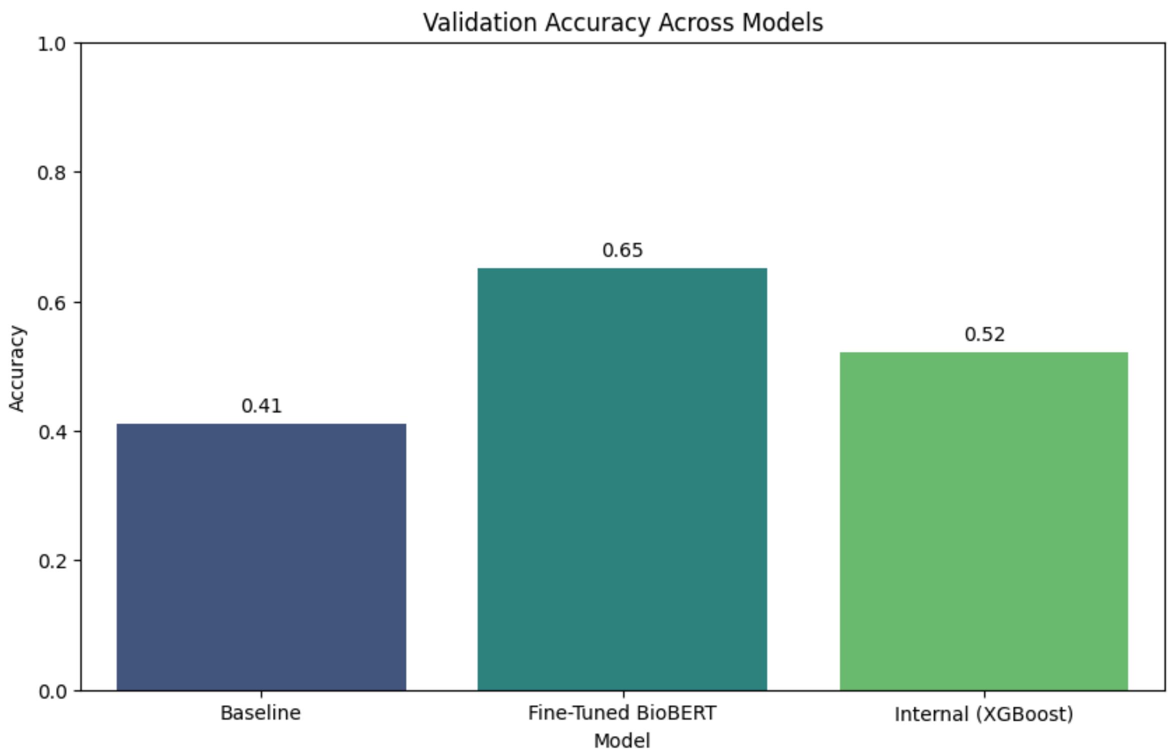
INCORPORATING LM OUTPUT :

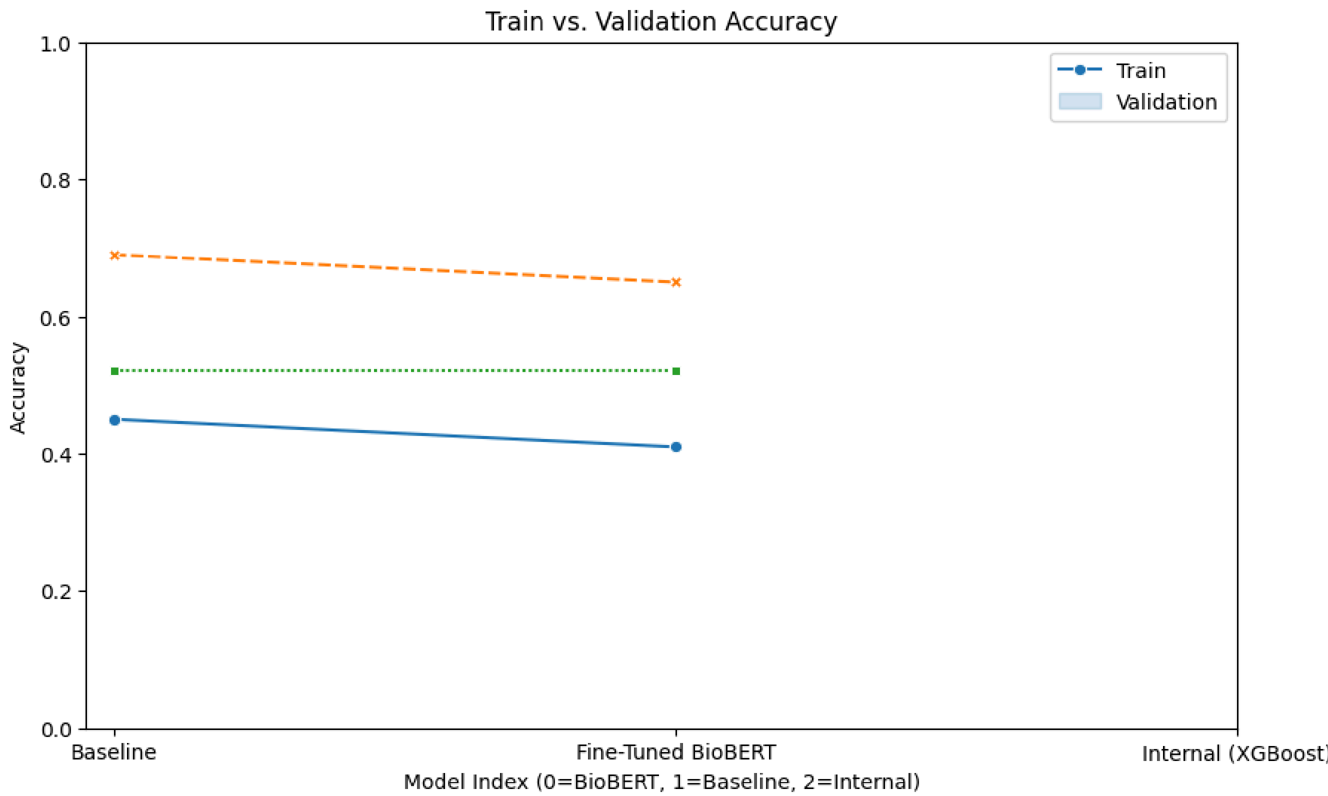
```
Internal LM (BioBERT Embeddings + XGBoost) - Accuracy: 0.1137, Macro F1: 0.0868
Internal LM Classification Report:
                                precision    recall  f1-score   support

         Allergy / Immunology       0.00      0.00      0.00         1
                      Autopsy       1.00      0.50      0.67         2
                   Bariatrics       0.00      0.00      0.00         4
   Cardiovascular / Pulmonary       0.21      0.23      0.22        74
                 Chiropractic       0.00      0.00      0.00         3
   Consult - History and Phy.       0.08      0.08      0.08       103
   Cosmetic / Plastic Surgery       0.00      0.00      0.00         5
                    Dentistry       0.00      0.00      0.00         5
                  Dermatology       0.00      0.00      0.00         6
         Diets and Nutritions       0.00      0.00      0.00         2
            Discharge Summary       0.21      0.14      0.17        22
         ENT - Otolaryngology       0.11      0.11      0.11        19
       Emergency Room Reports       0.00      0.00      0.00        15
                Endocrinology       0.00      0.00      0.00         4
             Gastroenterology       0.11      0.16      0.13        45
             General Medicine       0.07      0.06      0.06        52
         Hematology - Oncology       0.00      0.00      0.00        18
    Hospice - Palliative Care       0.00      0.00      0.00         1
       IME-QME-Work Comp etc.       0.00      0.00      0.00         3
     Lab Medicine - Pathology       0.00      0.00      0.00         1
                      Letters       0.00      0.00      0.00         5
                   Nephrology       0.00      0.00      0.00        16
                    Neurology       0.05      0.04      0.05        45
                 Neurosurgery       0.00      0.00      0.00        19
      Obstetrics / Gynecology       0.14      0.19      0.16        31
                 Office Notes       0.00      0.00      0.00        10
                Ophthalmology       0.17      0.12      0.14        17
                   Orthopedic       0.07      0.08      0.08        71
              Pain Management       0.56      0.75      0.64        12
        Pediatrics - Neonatal       0.05      0.07      0.06        14
    Physical Medicine - Rehab       0.00      0.00      0.00         4
```

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.

EDA ON TRAIN AND TEST RESULTS:




Train vs. Validation Accuracy
1.0
0.8
0.6
0.4
0.2
0.0
Accuracy
Train
Validation
Baseline
Fine-Tuned BioBERT
Internal (XGBoost)
Model Index (0=BioBERT, 1=Baseline, 2=Internal)