Automating Medical Billing Code Generation Using NLP

Tony Cen Cen[1], Andrew Juang[1]
Computer Science, Boston University, Boston, MA, USA[1]

Introduction

Medical coding is a critical step in healthcare administration where clinical documentation is translated into standardized billing codes like ICD-9 or ICD-10. This process enables reimbursement, tracks patient outcomes, and supports public health data collection.

However, medical coding is often manual, error-prone, and time-consuming. Coders must interpret complex clinical notes, which vary in structure, terminology, and completeness. Mistakes can result in denied claims, compliance issues, or delayed care.

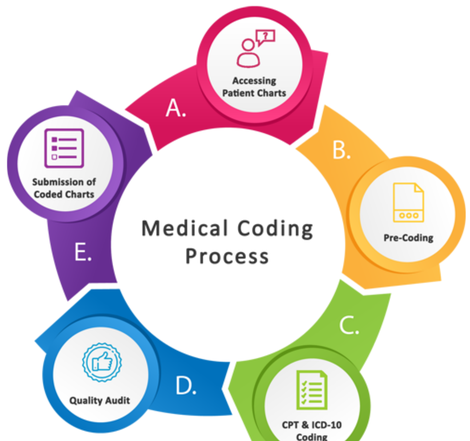
Our project addresses this challenge using natural language processing (NLP). We aim to build a system that automatically predicts appropriate ICD-9 codes from free-text clinical notes. By training models on the MIMIC-III dataset, which includes real ICU notes and ground truth codes, we hope to reduce administrative burden while maintaining high coding accuracy.

Motivation

This project was inspired by a conversation with a friend who worked at a medical insurance company. They described how they were talking to their boss about much time was spent reviewing and verifying billing codes submitted by clinics—and how often errors caused delays or rejections.

We realized much of this manual effort could be automated. With recent advances in NLP, it's now possible to train models that understand clinical text and assign appropriate billing codes, reducing both human error and administrative burden.

By using real ICU notes from the MIMIC-III dataset, our system learns to map free-text documentation to ICD-9 codes, improving coding speed and accuracy. This could help healthcare providers focus more on patient care and less on paperwork.



Methodology

We built an NLP pipeline to assign ICD-9 codes from clinical notes using the MIMIC-III dataset, which contains over 2 million ICU records. Our method involves several specialized stages tailored to the challenges of medical text.

1. Data Preparation:
We filtered for discharge summaries and joined them with primary diagnoses using HADM IDs. We removed de-identification markers (e.g., "[**Patient Name**]"), administrative metadata, and irrelevant headers, while preserving crucial patterns like "81-year-old", "125 mg IV", and "7.2 mg/dL" using placeholder tokens.

2. Text Preprocessing:

- **Tokenization:** Used NLTK's word tokenizer customized with regex for medical terms
- **Stopword Removal:** Applied NLTK stopword filtering but preserved medically important terms like "no", "not", "acute", "chronic"
- **Lemmatization:** Used WordNet lemmatizer, excluding protected clinical terms to avoid altering meaning
- **Abbreviation Expansion:** Converted terms like "HTN" to "hypertension" and "COPD" to "chronic obstructive pulmonary disease"
- **Section Tagging:** Replaced section headers (e.g., "HISTORY OF PRESENT ILLNESS") with semantic markers

3. Feature Engineering:
We created TF-IDF features using unigrams and bigrams, emphasizing rare but medically relevant terms. For deep learning, we embedded text using trainable word vectors tailored to the clinical domain.

4. Model Training:

- **Logistic Regression:** Trained with the 'saga' solver and L1 regularization, using class weighting to handle label imbalance
- **CNN:** Used 128 filters over 5-token windows with global max pooling and dropout regularization (embedding dim = 100, batch size = 64, seq len = 500)
- **BiLSTM:** Used bidirectional 64-unit layers to capture long-range dependencies (embedding dim = 200, batch size = 32, seq len = 1000)

5. Label Simplification:
We implemented hierarchical code grouping. Rare codes (fewer than 5 samples) were mapped to their parent categories (e.g., "428.21" → "428"). Increasing the grouping threshold to 10 improved accuracy from 9.59% to 13.36%.

This pipeline allowed us to model complex medical text while maintaining clinical interpretability and handling class imbalance effectively.

Evaluation Metrics

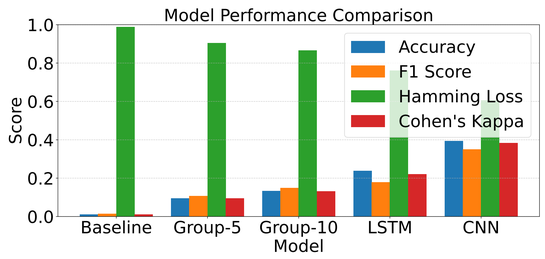
We used multiple metrics to assess model effectiveness:

- **Accuracy:** Fraction of notes where the top predicted code exactly matched the true ICD-9 code.
- **F1 Score (weighted):** Harmonic mean of precision and recall, weighted by class frequency—important for handling imbalanced medical datasets.
- **Hamming Loss:** Measures the fraction of wrong labels. In medical billing, a high Hamming Loss indicates frequent misclassifications, which could cause claim denials or billing errors.
- **Cohen's Kappa:** Measures the level of agreement between predictions and ground truth, correcting for random chance. Higher Kappa means our model adds real predictive value over guessing.

Performance Summary:

- Baseline Logistic Regression: Accuracy 1.16%, F1 Score 1.53%, Hamming Loss 98.84%, Cohen's Kappa 0.0114
- Statistical Grouping (min-count=5): Accuracy 9.59%, F1 Score 10.73%, Hamming Loss 90.41%, Cohen's Kappa 0.0952
- Statistical Grouping (min-count=10, 200 iterations): Accuracy 13.36%, F1 Score 14.89%, Hamming Loss 86.64%, Cohen's Kappa 0.1321

Model Performance Plot



Deep Model Results

CNN Model: Achieved 39.45% accuracy and 35.01% weighted F1 score. The CNN leveraged 128 filters over 5-token windows with global max pooling, trained using embeddings of dimension 100 and a sequence length of 500.

BiLSTM Model: Achieved 23.82% accuracy and 18.00% weighted F1 score. The bidirectional LSTM processed sequences of length 1000 with embedding dimension 200.

Why did CNN outperform LSTM? Although LSTMs are theoretically better at capturing long dependencies, in our task:

- Critical diagnostic information tends to appear in specific sections (e.g., "Discharge Diagnosis"), reducing the need for remembering long-term context.
- CNNs excel at detecting key phrases and localized patterns (e.g., "myocardial infarction"), which are sufficient to predict ICD-9 codes.

Hierarchical Code Grouping: Improved performance across all models by consolidating rare ICD-9 codes into parent categories. With a minimum count of 10, accuracy improved from 9.59% to 13.36% for logistic regression.

Conclusion

Our experiments show that NLP can assist in automating ICD-9 code assignment from clinical notes, helping reduce administrative overhead in medical billing. By applying TF-IDF vectorization and deep learning models to the MIMIC-III dataset, we demonstrated significant gains over traditional approaches.

Highlights:

- CNNs achieved 39.45% accuracy—over triple the baseline logistic regression's 1.16%.
- Statistical grouping of rare ICD codes helped stabilize training and boost performance.
- Preprocessing (tokenization, abbreviation expansion, section tagging) was critical to achieving high model performance.

While promising, the models still struggle with rare and ambiguous codes, and our current setup only supports single-label prediction. These limitations suggest directions for future work.

Future Work:

- Fine-tune domain-aware models like BioBERT or ClinicalBERT for better contextual understanding.
- Incorporate CPT (procedure) code prediction alongside diagnostic codes.
- Design a clinician-facing prototype for real-world deployment and feedback.