CS598 DEEP LEARNING FOR HEALTHCARE FINAL PROJECT PRESENTATION

PROJECT 17: **HICU: HIERARCHY FOR CURRICULUM LEARNING IN AUTOMATED ICD CODING**

TEAM 26 (RATULS2 / RATULS2@ILLINOIS.EDU)

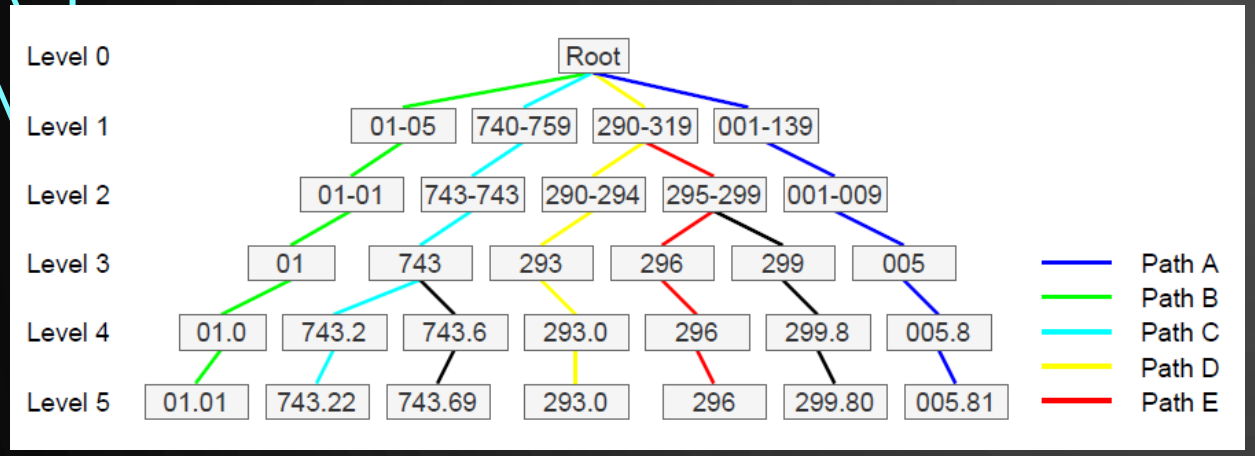
AGENDA

- Introduction / Description of the Problem
- HiCU Algorithm, Architecture, Originals Approach
- My Hypothesis / Claim
- Environment Setup, Code Base, Data Preparation
- Result Reproduction Attempts
- Discussion on Study Results
- Conclusion

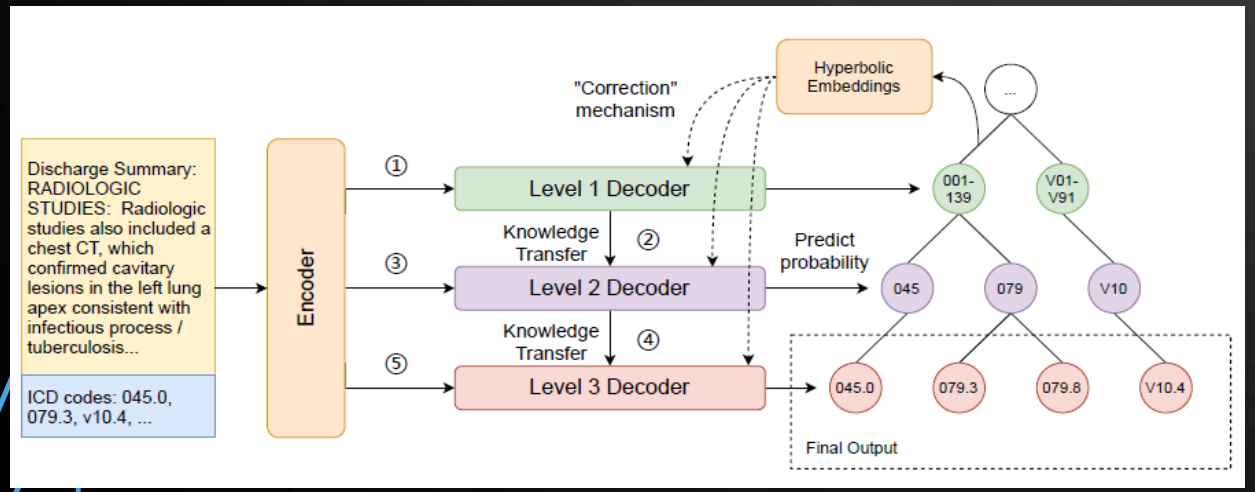
INTRODUCTION / DESCRIPTION OF THE PROBLEM

- Paper Title: "HiCu: Leveraging Hierarchy for Curriculum Learning in Automated ICD Coding“ by Ren, Zeng, Zhu, Krishnan.
- The Problem: ICD codes are globally used for coding various diagnoses, symptoms, and procedures documentation in healthcare. Coding involves the classification of textual clinical notes, discharge summaries, and patient profiles to ICD codes. Coding task is complex, time-consuming and error prone. The automation is aimed at improving efficiency and improve medical documentation.
- Traditional CNNs, RNNs, and transformer models treat each code as an independent label, leading inefficiencies and inaccuracies, specially with rare codes. Hierarchical and imbalanced nature of the codes is not leveraged.
- HiCu / Hierarchical Curriculum learning technique leverages the hierarchical structure of the ICD codes to learn complex relationships between codes.

HICU ALGORITHM, ARCHITECTURE, APPROACH



A subset of ICD code level tree is shown as example



HiCu Architecture

- Encoder-Decoder framework with hierarchical learning
- Sequential Training Algorithm – learning from simple to more complex codes (first level label tree codes are learned through first level encoders). Then proceed to higher levels through knowledge transfer mechanism.
- Hyperbolic Embeddings are used at each level to guide the attention computation.
- Uses a High-order Grouping of ICD Code Blocks to create a 2-level hierarchy.

HYPOTHESIS

Authors used three encoder architectures to test the effectiveness of HiCu for automated ICD coding: The Bidirectional Long-Short Term Memory (Bi-LSTM) based on LAAT; Multi-Filter Residual Convolutional Neural Network (MultiResCNN) model designed on TextCNN and ResNet; RAC reader with Convolved Embedding Module and the Self-Attention Module.

Model	AUC		F1		Precision@K		
	Macro	Micro	Macro	Micro	P@5	P@8	P@15
CAML	89.5	98.6	8.8	53.9	-	70.9	56.1
DR-CAML	89.7	98.5	8.6	52.9	-	69.0	54.8
MSATT-KG	91.0	**99.2**	9.0	55.3	-	72.8	58.1
HyperCore	93.0	98.9	9.0	55.1	-	72.2	57.9
JointLAAT	92.1	98.8	10.7	57.5	80.6	73.5	59.0
LAAT*	92.0±0.11	98.8±0.02	9.7±0.24	57.4±0.16	81.2±0.22	73.9±0.17	59.0±0.14
w/ HiCuA	**94.8**±0.07	99.1±0.01	10.2±0.21	57.4±0.11	81.2±0.12	73.9±0.10	59.1±0.09
RAC*	93.0±0.08	98.8±0.02	7.9±0.30	55.4±0.27	80.8±0.15	73.2±0.18	57.8±0.11
w/ HiCuA	94.3±0.09	99.0±0.01	8.4±0.17	56.5±0.17	81.2±0.32	73.8±0.17	58.8±0.12
w/ HiCuC	94.4±0.15	99.0±0.01	8.4±0.54	55.8±0.44	81.1±0.22	73.6±0.20	58.6±0.12
MultiResCNN*	91.2±0.23	98.7±0.02	8.6±0.40	56.2±0.34	81.7±0.17	74.3±0.20	59.1±0.22
w/ HiCuA	94.7±0.10	99.1±0.02	9.2±0.33	56.7±0.29	82.0±0.14	74.8±0.16	59.6±0.07
w/ HiCuC	94.6±0.12	99.1±0.01	9.3±0.55	56.6±0.45	82.1±0.11	74.8±0.17	59.6±0.14
w/ HiCuA+ASL	93.7±0.20	98.9±0.02	11.4±0.36	**57.6**±0.13	**82.4**±0.16	**75.1**±0.14	**59.8**±0.12
w/ HiCuC+ASL	94.0±0.33	98.9±0.04	**11.5**±0.41	57.4±0.21	**82.4**±0.25	**75.1**±0.19	59.7±0.09

During the Knowledge transfer process, authors proposed 2 methods in Hyperbolic embedding to generate query vectors: Addition (HiCuA) and Concatenation (HiCuC).

My hypothesis is to prove the below claim through reproduction of the model performance results:

HiCuA (Hyperbolic Correction Addition) significantly enhances the MultiResCNN model's performance (improved AUC and F1 scores).

ENVIRONMENT SETUP, DATA PREP

- Download MIMIC-III data from its original secure store
- Setup two environments:
  - Local windows computer: for running the pre-processing steps. The processed datasets were loaded into Google Drive.
  - Google Colab Pro based cloud environment trains various models on pre-processed datasets to generates model performances.
- After each training model is run, the model performance parameters are captures and compared with findings from Author's original study.

MODEL TRAINING AND REPRODUCTION ATTEMPTS

- Objective: re-establish hierarchical learning using HiCu algorithm, which is more efficient than non-hierarchical learning on ICD codes.
- I did the training on MultiResCNN model only. Due to time-constraint and hardware limitations REC and LAAT model executions could not be complete.
- Reproduction attempts include:
  - Changing of hyper parameters to compute the results in my environment.
  - Choosing available hardware due to demand of high CPU, GPU and RAM
  - Code changes needed for latest version of Python and other packages.
  - Run original hierarchical learning algorithm and capture metrics (AUC, F1, Loss, Precision etc.)
  - Modifying original code to flatten ICD-hierarchy (ablation study), train the models and capture necessary metrics for comparison.

RESULT DISCUSSION

Original claim (Authors' paper)

Model	AUC Macro	AUC Micro	Precision @8
MultiResCNN	91.2	98.7	74.3
MultiResCNN with HiCuA	94.7	99.1	74.8

Claim 1 (results in a different environment) without code changes

Model	AUC Macro	AUC Micro	Precision @8
MultiResCNN with HiCuA	94.23	99.06	83.09

My metrics are in the similar range as the authors

Claim 2 (results on ablation study) with code changes on flatten hierarchy

Model	AUC Macro	AUC Micro	Precision @8
MultiResCNN	89.83	92.66	50.95
MultiResCNN with HiCuA	91.15	93.24	52.35

My metrics are significantly lower than the authors

CONCLUSION

- Reaffirming the significance and effectiveness of the hierarchical learning algorithm.
- HiCu approach is more innovative and efficient than traditional non-hierarchical / flat learning approach.
- HiCu algorithm is based on the hierarchical nature of the ICD codes, that results into better performance.