Automating Key Phrases identification in Patient Notes

Group 8: Ujwala Bayana, Abhishek Andluru, Harshitha Somala, Harshitha Yeddula, Rajesh K

1. Description

When you visit a doctor, how they interpret your symptoms can determine whether your diagnosis is accurate. By the time they're licensed, physicians have had a lot of practice writing patient notes that document the history of the patient's complaint, physical exam findings, possible diagnoses, and follow-up care. But recently, the Step 2 Clinical Skills examination became a component of the United States Medical Licensing Examination® (USMLE®). The exam required test-takers to interact with Standardized Patients (people trained to portray specific clinical cases) and write a patient note. Trained physician raters later scored patient notes with rubrics that outlined each case's important concepts (referred to as features). The more such features found in a patient note, the higher the score (among other factors that contribute to the final score for the exam). However, having physicians score patient note exams requires significant time and human resources.

We will strive to discover an automated method to translate clinical ideas from the test rubric to many ways these concepts are conveyed in the patient clinical note in this project, which will allow a professional doctor to spend less time evaluating the notes of medical students and interns.

2. Dataset

The data was obtained straight from kaggle, with the entire zip file including four files, namely features. csv - features rubric for each clinical case, patient notes.csv - history of patient notes, train.csv, and test.csv

We clean the data first and then merge all the files into a single dataset file because the data is so raw. Our merged dataset contains 14, 300 datapoints where each data point contains history notes of a patient, the feature we are looking for and different locations of our feature text in the history notes. Example of patient notes with feature locations:

	id	case_num	pn_num	feature_num	annotation	location	feature_text	pn_history
0	00016_000	0	16	0	[dad with recent heart attcak]	[696 724]	Family-history-of-MI-OR-Family-history-of-myoc...	HPI: 17yo M presents with palpitations. Patien...
1	00016_001	0	16	1	[mom with "thyroid disease]	[668 693]	Family-history-of-thyroid-disorder	HPI: 17yo M presents with palpitations. Patien...
2	00016_002	0	16	2	[chest pressure]	[203 217]	Chest-pressure	HPI: 17yo M presents with palpitations. Patien...
3	00016_003	0	16	3	[intermittent episodes, episode]	[70 91, 176 183]	Intermittent-symptoms	HPI: 17yo M presents with palpitations. Patien...
4	00016_004	0	16	4	[felt as if he were going to pass out]	[222 258]	Lightheaded	HPI: 17yo M presents with palpitations. Patien...

3. **Methodology and Expected Results**

   Annotation or required feature (e.g., “diminished appetite”) will be expressed in various ways in the clinical notes written by students (e.g., “eating less,” “clothes fit looser”). We expect our model to output locations for each Annotation in the patient history (pn_history) of the validation set.

   We will build several different models and compare the accuracy results of each model. Since there are many options for our models, we will try various approaches and identify which ones are in the scope of this project. We will definitely build a Neural Network model and BERT (Bidirectional Encoder Representations from Transformers). We are also interested in advancements of BERT like RoBERTA, DeBERTa and BioBERT but we are concerned about the computation expenses and complexity of these algorithms. An alternative to this is to build a LSTM neural network, which will have less power in processing word sequences.

4. **Ablation Study and Evaluation Metrics:**

   As for Performance Evaluations we are planning to use cross validation techniques, Confusion matrix and F1 score.

   Ablation study: Improvements with F-1 scores and model loss of BERT compared to a simple Neural Network base model.

5. **Timeline**
   1. **Week 9,10:** Learn PyTorch, understand various Transformer-based techniques
   2. **Week 11:** Build a Neural Network model for our dataset and BERT
   3. **Week 12:** Testing output, compare different models and make some visualizations as results
   4. **Week 13:** Write the report

6. **Related Work**

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova (2018) BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

[2] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov (2019) RoBERTa: A Robustly Optimized BERT Pretraining Approach

[3] Pengcheng He, Xiaodong Liu, Jianfeng Gao, Weizhu Chen (2021) DEBERTA: Decoding-Enhanced Bert With Disentangled Attention

[4] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang (2019) *BioBERT:* A Pre-trained Biomedical Language Representation Model For Biomedical Text Mining.