Automated Scoring of Clinical Patient Notes: Findings From the Kaggle Competition and Their Translation into Practice

Victoria Yaneva[1], King Yiu Suen[1], Le An Ha[2], Janet Mee[1], Milton Quranda[1], and Polina Harik[1]

[1]National Board of Medical Examiners, Philadelphia, USA
{vyaneva, ksuen-temp,jmee, mquranda, pharik} @nbme.org
[2]Ho Chi Minh City University of Foreign Languages, Vietnam
anhl@huflit.edu.vn

Abstract

Scoring clinical patient notes (PNs) written by medical students is a necessary but resource-intensive task in medical education. This paper describes the organization and key lessons from a Kaggle competition on automated scoring of such notes. 1,471 teams took part in the competition and developed an extensive, publicly available code repository of varying solutions evaluated over the first public dataset for this task. The most successful approaches from this community effort are described and utilized in the development of a PN scoring system. We discuss the choice of models and system architecture with a view to operational use and scalability, and evaluate its performance on both the public Kaggle data (10 clinical cases, 43,985 PNs) and an extended internal dataset (178 clinical cases, 6,940 PNs). The results show that the system significantly outperforms a state-of-the-art existing tool for PN scoring and that task-adaptive pretraining using masked language modeling can be an effective approach even for small training samples.

1 Introduction

A core practice in assessing the clinical skills of medical students is the use of Objective Structured Clinical Examinations (OSCEs) – a type of exam, where test-takers interact with *standardized patients*, who are trained to portray a set of clinical symptoms. After examining the patients, the test-takers are asked to describe their findings in a clinical patient note (PN), similar to those found in electronic health records (see an example PN in Appendix A). The PN serves as a documentation of the encounter and is used to assess examinee ability to gather information, record physical examinations, and interpret clinical data. OSCEs are widely used in medical schools in various countries, with around 90% of US schools requiring their students to pass such exams (Barzansky and Etzel, 2016).

A major bottleneck for scaling OSCE assessment is the time, cost, and effort associated with the expert grading of large amounts of PNs, especially given limited faculty time. For example, in the former United States Medical Licensing Examination® (USMLE®) Step 2 Clinical Skills exam (discontinued in 2020), more than 100 licensed physician raters were needed every year to grade ≈ 330,000 PNs from ≈ 35,000 US and international test-takers (Sarker et al., 2019).

While there is interest among medical educators to address the above limitations using automated grading methods, the exploration of such methods has been slow and fragmented due to exam security concerns, which limit data sharing. This has resulted in small-scale, predominantly internal explorations of automated scoring, with no shared datasets or code to foster collaborative research.

To address this gap, we organized a Kaggle competition on clinical PN scoring[1] as a community effort to move this field forward. We then used the most successful approaches for the development of an interpretable and transparent PN scoring system. The contributions of this paper are as follows:

- Description of the Kaggle competition on clinical PN scoring, for which we released a public dataset and which resulted in a large repository of publicly available code.
- Analysis of the most successful approaches.
- Description of an Amazon Web Services (AWS) proof-of-concept for PN scoring based on the successful solutions; Choice of models and system architecture are discussed with a view to operational scalability. Models are

[1]https://www.kaggle.com/c/nbme-score-clinical-patient-notes/overview

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 87–98

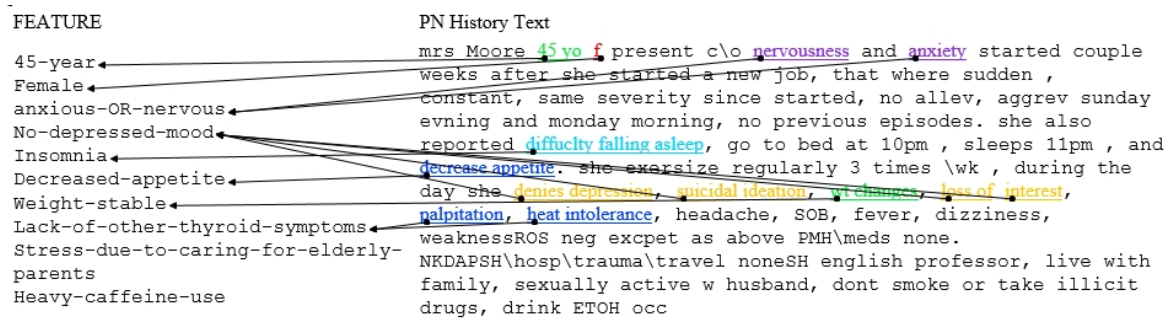

Figure 1: Example of rubric features and their annotated expressions within a patient note excerpt

trained in a real-world scenario of limited PNs for newly developed cases.

- Baseline comparison to an already operationalized scoring system with a mean F1 score improvement from .76 to .95. Performance is evaluated on both the Kaggle data and an extended internal dataset.
- Evaluation of a scenario when training is performed using limited annotation.
- Discussion of ethical considerations and implications for fairness, reliability, and validity.

2 Context

The data used in this study originated from the United States Medical Licensing Examination® (USMLE®) – a series of examinations used to support medical licensure decisions in the United States. Until 2020, the USMLE Step 2 Clinical Skills examination was a part of the USMLE step sequence and involved interactions with standardized patients portraying different clinical scenarios. The resulting PNs were graded using rubrics specific to each clinical case, which contain a set of *features* – important concepts, which should appear in an appropriately documented PN (Figure 1). For example, for a clinical case about a patient with anxiety, it may be important that the examinee discovers that the patient has *insomnia*, in which case *insomnia* would be listed as a rubric feature. PNs that do not mention that symptom or some expression of it such as *difficulty falling asleep* would receive a lower rater score.

Key challenge for automated scoring is the variety of ways features are expressed (e.g., *evaluation for coronary risk factors* expressed as *father with MI at age 50*, or *denies depressed mood* expressed as *(-) anhedonia*). There are cases of ambiguous negation as in *denies nausea, vomiting* for the feature *no nausea and or vomiting* or temporal aspects such as *recent URI* for *uri one week ago*. To be operationally usable, a PN scoring system needs to provide interpretable evidence and be highly accurate. These requirements are crucial to ensure exam fairness and protect the health of the public.

3 Related Work

The vast majority of work on automated scoring has been done in the field of writing evaluation (see Klebanov and Madnani (2020) for an overview). Studies on scoring clinical text include Latifi et al. (2016), who use a feature-based system for scoring short responses to clinical decision-making questions, Ha et al. (2020) who predict examinee proficiency from responses to clinical short-answer questions, and Suen et al. (2023) who use transformer models for scoring short answers to clinical questions. For PN scoring specifically, Yim et al. (2019) use features and BERT embeddings for scoring a corpus of 338 PNs and Zhou et al. (2022) use weakly supervised approaches and transfer learning for scoring two clinical cases of 30 PNs each.

The work most relevant to ours is the INCITE system (Sarker et al., 2019; Harik et al., 2023), which was developed for operational scoring of PNs from the USMLE Step 2 CS exam and which we use as a baseline. The system is a modular pipeline which outputs a binary score of "found" or "not found" for each rubric feature, utilizing custom-built lexicons and annotations. The first two modules perform direct and fuzzy matching between a feature or a lexicon variant and the PN text using a fixed or dynamic Levenshtein ratio threshold. Any features whose expressions are found using this method are removed from the pipeline to optimize running time. Next, matching is performed against combinations of lexicon variants and annotations, which "often leads to an explosion of the number of eventual entries" (Sarker et al., 2019) as terms in the annotations are replaced with

variants from the lexicons. To limit this search space, there is a cap of 10,000 randomly sampled combinations per feature. Matching is then done using these phrases as sequences and as bag-of-words to cope with fragmented entries [2].

Advantages of the INCITE system include its high performance, ability to be tuned for precision and recall by varying the thresholds, as well as its speed – it is capable of processing over 50,000 PNs per day on a desktop computer. However, the rule-based nature of the system limits improvement from more training data, especially because more annotations would greatly increase the search space for supervised concept detection.

4 Task description and evaluation

The task of developing an interpretable system for automated scoring of PNs is one where features from the rubric are mapped to expressions from the PN. If an expression of the feature is identified in the PN then the feature is considered "found", else it is "not found". The more features are found, the higher the score for that PN. We perform two types of model evaluation, as described below.

Token-level evaluation: This type of evaluation answers the question "What phrase spans in the PN correspond to a given rubric feature?". This evaluation is identical to the one used in the Kaggle competition and comparable to its leaderboard.

For each instance, the system predicts a set of character spans that it considers to correspond to that feature, where a character span is a pair of indexes representing a range of characters within a text. These predicted spans are then compared to ground-truth spans from the annotation and scored as: a character is considered true positive if it is within both a ground-truth and a prediction; false negative if it is within a ground-truth but not a prediction; and false positive if it is within a prediction but not a ground truth. An overall F1 score is computed from the TPs, FNs, and FPs aggregated across all instances[3].

Binary evaluation: This type of evaluation answers the question "Was an expression of a feature found (1) or not found (0) in the PN?". This evaluation corresponds to the way PNs are scored in practice.[4] If at least one span is identified as corresponding to the feature, the feature is considered "found". For the neural models, binary scores are obtained by applying a function over the token-level predictions using a threshold of 0.5.

5 Data

Training and evaluation are performed in two datasets of PN history portions[5] – public and proprietary – from the USMLE Step 2 CS exam.

Public dataset: This dataset was used in the Kaggle competition (so henceforth referred to as "the Kaggle dataset") and contains the history portions of 43,985 PNs from 10 clinical cases and the corresponding features for each case. Data were collected between 2017 and 2020 from 35,156 US or international test-takers who took the exam under standardized conditions in one of five testing locations in the US. The average number of PNs per case is 4,398 (min = 992, max = 9,936), total number of tokens is 5,958,464, and the average length of each history portion is 135.47 tokens (SD = 24.27). The average number of features per history portion is 14.3 (SD = 3.34). Of these, a total of 2,840 PNs (284 per case) were annotated by 10 experienced US medical practitioners who were asked to identify the spans of each phrase that is an expression of a rubric feature and link it to that feature. The annotators were divided in pairs of two and 20% of the PNs from each case were double-rated (see detailed annotation guidelines and procedure in Appendix B). F1 agreement scores were computed using the token-level evaluation procedure described above and showed a substantial agreement across all cases (F1 = .84 (SD = 0.075); Cohen's κ of 0.89 (SD = 0.057)). Binary F1 denoting whether an expression of a given feature was found in a PN was F1 = 0.97 (SD = 0.014). Detailed information about the corpus can be found in Yaneva et al. (2022). The data is available via a data sharing agreement at https://www.nbme.org/services/data-sharing.

Proprietary dataset: This dataset consists of a much larger number of clinical cases – 178 – with

[2] E.g., "Antibiotics taken in recent times for his symptoms – negative". As Sarker et al. (2019) note, window-based fuzzy matching would fail to include the negation and the rest of the description in one window.

[3] For specific examples, see https://www.kaggle.com/competitions/nbme-score-clinical-patient-notes/overview/evaluation

[4] Raters are not typically required to mark the exact expressions that correspond to a feature. As a result, human scores are not explicitly traceable to specific evidence in the PN, unless this is specifically required (e.g., if a score is contested).

[5] The history portion is where all relevant clinical information obtained from an interview with the patient is described.

fewer PN history portions per case (μ = 39; min = 32; max = 43). Total number of PNs in the set is 6,940, and total number of tokens is 1,121,236. Average document length is 161.56 tokens (SD = 29.42), and the average number of features is 13.92 (SD = 4.84). All PNs were annotated following the same procedure as above, resulting in binary inter-annotator agreement of F1 = .95 (SD = 0.09), computed over 10% double-rated notes per case.

6 Kaggle competition: Key lessons

The Kaggle competition on scoring clinical PNs resulted in a total of 28,049 code entries from 1,471 participating teams. After the end of the competition, many teams posted their notebooks in the competition's code repository, making them publicly available. In this section, we analyze the top 15 publicly shared solutions [6] (the teams ranking from 1st to 11th place, and those that ranked #13, #14, #18, #19, and #20), as well as insights from other notebooks and key forum discussions. The final leaderboard rankings can be seen at https://tinyurl.com/p9mwfu8c and corresponding code contributions can be accessed at https://tinyurl.com/3h8p5a67.

Results Many of the top-performing teams reached a token-level F1 score of .89, with minor differences between solutions (e.g., #1 F1 = .89456, #2 F1 = .89432, and #3 F1 = .89384), indicating that there are different, equally successful ways of addressing this task. This result also suggests potential ceiling effects arising from annotation inconsistencies such as not capturing every instance of a phrase that can be mapped to a feature[7] or not identifying the correct character span of a phrase (average inter-annotator agreement F1 = .84). Such inconsistencies resulting from human error are inevitable in spite of rigorous training and data cleaning efforts, further showcasing the need for improved reliability in scoring.

Key approaches Most high-performing solutions used some version of DeBERTa (He et al., 2021) as the backbone and performed **task-adaptive pretraining** (Gururangan et al., 2020) by using masked language modeling (MLM) over the unannotated portion of the data. One solution (#2) additionally pretrained on the SQuAD 2.0 question answering dataset (Rajpurkar et al., 2018), drawing a parallel between the two tasks: the feature text in PN scoring corresponds to the *question* in the SQuAD data, the patient history is the *context*, and the annotations are all *answers* to the question.

Another approach shared by almost all of the analyzed solutions was the use of **pseudo labeling** (Arazo et al., 2020) to create more training data from the unannotated notes. One team (ranking #3) also utilized **meta pseudo labeling** (Pham et al., 2021). Some teams reported that hard labels work better than soft labels [solutions ranking #8, #70], while others reported the opposite [#1].

While these techniques were used in most high-performing solutions, one approach that distinguished the Top 3 winners was the use use of **multi-task learning**. In this case, the main task of token classification is combined with an auxiliary task of predicting annotation span boundaries, putting more weight on tokens that are the beginning or end of a phrase. In the model architecture, this is expressed as a primary head for token classification and two auxiliary heads for span boundary detection (one for starts and one for ends).

A focal point for most successful solutions was the prevention of overfitting. This was done through careful ensembling and detailed experimentation with various dropout rates, as well as extensive use of cross validation.

7 Models

Two key differences between real-world applications and the competition are that: i) newly developed cases do not come with large amounts of unannotated PNs (which makes pseudo-labeling not suitable), and ii) the trade-off between performance gain and resource requirements such as speed and compute power is an important aspect of model selection (making the ensembling of a large number of models impractical). With these prerequisites in mind, the following approaches were trained and evaluated.

INCITE baseline: The INCITE system is an operationally used benchmark. The case-specific data in its lexicons is from the training set for each case.

DeBERTa baseline: The pretrained DeBERTa v3 (He et al., 2021) was used as the backbone

[6] Detailed solution descriptions for first place: https://tinyurl.com/2p8afa94, second place: https://tinyurl.com/yc77s4rk, and third place https://tinyurl.com/3yf4u6hr.

[7] The 2nd place winner hypothesised that annotators were more likely to miss repeated annotations than first occurrence and noted that the use of recursive neural networks (RNNs) could be useful to capture such sequence dependencies.

Token-level results for the public (Kaggle) dataset

	Public test set			Private test set		
	P	**R**	**F1**	**P**	**R**	**F1**
DB	**.846**	.882	.864	.85	.885	.867
DB + MTL	.844	.882	.862	**.849**	.887	.868
DB + MLM	.845	**.889**	**.866**	**.849**	**.89**	**.869**

Token-level results for the proprietary data (178 cases)

	Training set (80%)			Test set (20%)		
	P	**R**	**F1**	**P**	**R**	**F1**
DB	.836	.782	.808	.681	.768	.722
DB + MTL	.845	.808	.826	.773	.7	.745
DB + MLM	**.94**	**.95**	**.945**	**.856**	**.834**	**.845**

Table 1: Token-level results. DB = DeBERTa; MTL = multi-task learning; MLM = masked language modeling; P = precision, R = recall. Note that INCITE does not output token-level information.

model[8], where each token was assigned a label of 1 if inside the annotation span and 0 otherwise. The output of the model was the probability of each token being inside the annotation span. After experimentation with various probability thresholds in both datasets, a threshold of 0.5 was determined sufficient (i.e., a token with a probability greater than 0.5 was considered to be inside the span). The model was trained with cross-entropy loss.

DeBERTa + Masked Language Modeling (MLM): 15% of the tokens in the input sentences were randomly masked and ran through the model, where the model's objective was to predict the masked tokens. For the Kaggle dataset, the pretraining was performed on the unlabeled data. For the proprietary dataset, there were no unlabeled data, so the pretraining was performed on the labeled data from the training set. The MLM model was pretrained for one epoch. The pretrained model was then trained the same way as the baseline model.

DeBERTa + Multi-task Learning (MTL): Two auxiliary tasks were trained jointly with the model, predicting whether the token was at the beginning (Task 1) or the end (Task 2) of the annotation span.

8 Results

Token-level results: Table 1 presents the results from the token-level evaluation. For the Kaggle data, we kept the exact training, private test, and public test sets,[9] so the results are directly comparable to the competition leaderboard. As shown in the table, the best-performing model is DeBERTa + MLM, with a private test set F1 score of .869 (P = .849, R = .89). This compares to F1 = .89456 for the #1 Kaggle solution. A drop in performance of only .03 points shows that the exclusion of pseudo-labeling and the use of a single model instead of an ensemble of multiple models did not lead to a loss that has a practical significance (although such difference is important in a competition context).

For the internal dataset the results are consistent with Kaggle – the best model is again DeBERTa + MLM (F1 = .845, P = .856, R = .834). The model generalizes over a much larger set of cases and is robust when trained on fewer notes (as a reminder, the internal dataset contains 32 to 49 annotated notes per case (80% used for training), compared to 100 training notes per case in Kaggle). Importantly, this result shows that MLM pretraining can be fruitfully applied to small training sets, leading to an increase over the DeBERTa baseline (.845 vs. .722). The DeBERTa and DeBERTa + MTL results did not generalize as well, exemplifying the importance of task-adaptive pretraining.

Note that token-level evaluation was only performed with the neural models. INCITE cannot output specific phrases if the matching was done by some of its more advanced modules (e.g., bag of words from lexicon variants + fuzzy matching). This is an important distinction between INCITE and the neural approaches that has implications for both interpretability and intended use (e.g., in providing feedback to learners).

Binary evaluation results and comparison to INCITE: The binary evaluation results are presented in Table 2. For Kaggle, the neural models outperform INCITE (F1 of .958 for DeBERTa + MLM; .888 for INCITE on the public test set). This difference is more pronounced for the proprietary dataset, where DeBERTa + MLM's robust F1 of .952 compares to an F1 of .761 for INCITE and .946 for inter-annotator agreement. As shown, the main difference with INCITE is that DeBERTa + MLM has a much higher recall (e.g., R = .954 vs. R = .642 for INCITE). Precision is high for both DeBERTa + MLM (.95) and INCITE (.953).

The binary evaluation results on the internal

[8]Learning rate: 1e-4; Optimizer: AdamW; Weight decay: 0.01; Learning rate scheduler: Linear (warmup for 10% of the training steps); Training epochs: 5; Training batch size: 4 per device x 2 GPUs = 8; Gradient accumulation steps: 4.

[9]In Kaggle, the public test set serves as a validation set for the development of the approaches. The final leaderboard is determined after the end of the competition by the performance of the submitted models on the private test set.

Binary evaluation for the public (Kaggle) dataset

	Public test set			Private test set		
	P	**R**	**F1**	**P**	**R**	**F1**
INCITE	**.962**	.818	.883	**.961**	.828	.888
DB	.95	**.962**	.956	.951	**.963**	.957
DB + MTL	.947	.961	.954	.953	**.963**	**.958**
DB + MLM	.952	.961	**.957**	**.961**	.956	**.958**

Binary evaluation for the proprietary data (178 cases)

	Training set (80%)			Test set (20%)		
	P	**R**	**F1**	**P**	**R**	**F1**
INCITE	.966	0.85	.902	**.953**	.642	.761
DB	.927	.933	.93	.896	.862	.879
DB + MTL	.933	.947	.94	.898	.877	.888
DB + MLM	**.979**	**.979**	**.979**	.95	**.954**	**.952**

Table 2: Binary evaluation results. DB = DeBERTa; MTL = multi-task learning; MLM = masked language modeling, P =precision, R = recall.

dataset for DeBERTa + MLM (F1 = .952) are comparable to human-rater performance as computed on the set of double-rated PNs per case (inter-rater agreement F1 = .946).

Out of the total of 19,465 instances, INCITE and DeBERTa + MLM agreed in 14,532 or 75% of the instances (κ = 0.51, indicating moderate agreement); INCITE vs Annotation agreement was κ = 0.52; Finally, DeBERTa + MLM vs Annotation agreement was κ = 0.89.

Limited annotation setting: For the internal dataset we also experiment with a limited annotation setting, since the question of how much annotation is required before a model can be trained has strong practical implications. For a limited annotation scenario where we train on 30% of the data (i.e., $\approx$ 12 PNs per case) and evaluate on 70% held-out data, the F1 score for DeBERTa + MLM is .836 (binary F1 = .94) compared to .69 (binary F1 = .83) for DeBERTa + MTL and .64 (binary F1 = .86) for DeBERTa baseline. These results show that task-adaptive pretraining leads to robust models even in a limited annotation scenario.

9 Error Analysis

For DeBERTa + MLM, there were 990 errors (594 FNs and 396 FPs), distributed across all 178 clinical cases[10]. However, the errors were only distributed across 36% of the 1815 features. We hypothesize that this may be due to differences in annotation length for different features. Indeed, the mean annotation length differs between the correct predictions and the errors: it is μ = 19.6 (SD = 20.7) for correct and μ = 13.2 (SD = 17.2) for the errors (Mann-Whitney U = 7183226, $p < 0.001$). This is somewhat counter-intuitive, as it suggests that the shorter features and shorter annotations are more difficult to detect. Further content-specific analysis is needed to illuminate the potential causes for this phenomenon. Annotation length affected INCITE inversely and to a much greater extent, where the annotations for the correct class (μ = 15.9, SD = 19.8) are on average twice as short as the errors (μ = 29.12, SD = 19.9), (U = 19231079.5, p = 0.0), potentially due to limitations from its window-based approach. Spearman correlation between annotation length and correct/incorrect predictions further supports this finding: r = 0.08 for DeBERTa + MLM model and r = -0.36 for INCITE. A likely explanation for this result is that INCITE's window-based approach is challenged by long phrases, while DeBERTa's multi-head self-attention layers, where the encoder reads the entire sequence bidirectionally, enables it to cope well with these. In addition, since the objective of the neural models was to decide whether a given character belongs to a relevant phrase, the higher character count of longer phrases increases the available information for making a prediction. Further analysis of the differences between correct and erroneous predictions did not reveal a specific pattern. This extended analysis is presented in Appendix D together with examples of specific features.

10 Deployment

A system based on the DeBERTa + MLM model was deployed on the Amazon Web Services (AWS) platform. A graph depicting the AWS architecture can be seen in Appendix C. Figure 2 provides a visualization of the system output. Speed, resource efficiency, and scalability are ensured by the use of SageMaker and eliminating the need for human interference via event triggers: placing incoming data in an initial S3 bucket triggers a series of Lambda functions, which initiate preprocessing, training, and scoring.

11 Discussion

The results presented above showed that the best model, DeBERTa + MLM, led to significant im-

[10]The average number of errors per case was 10.5 (SD = 9.15), with 4 cases scored without any errors, 16 cases with one, and 22 cases with two errors; highest number of errors in a case was 18 (1 case), followed by 17 (1 case), and 15 (3 cases). The number of errors per case (μ = 10.5 (SD = 9.15)) was best explained by the number of features in a case, where cases with higher number of features had more errors.

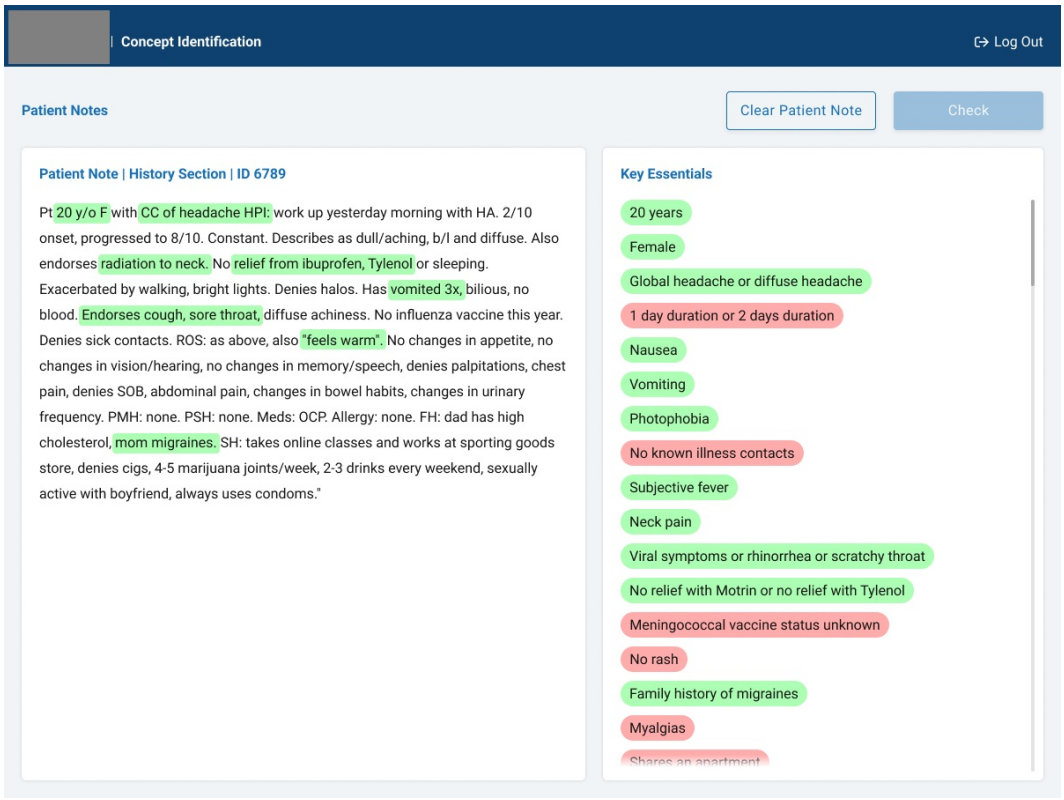

Figure 2: System output for an example PN

provements over INCITE for a diverse set of 178 clinical cases (binary F1 = .95 for DeBERTa + MLM compared to .76 for INCITE), as well as the Kaggle data (.96 vs .89). INCITE was significantly more challenged by lengthy phrases and the smaller number of training instances in the proprietary dataset. By contrast, as shown when evaluating in the limited annotation scenario, DeBERTa + MLM continues to yield meaningful gains when trained on as few as 12 PNs. These experiments add evidence that task-adaptive pretraining can be beneficial even for small training samples, making the approach applicable to a wide range of practical scenarios.

While the INCITE system struggled to identify lengthy expressions (i.e., the annotations of the errors were twice as long as those of the correctly identified instances), the DeBERTa + MLM model coped well with long sequences. This is likely due to the multi-head self attention layers of DeBERTa, where the encoder reads the entire sequence in a bidirectional manner. In addition, since the task was to decide whether a given character belongs to a relevant phrase or not, the higher character count of longer phrases increases the available information for making a prediction. At the same time, INCITE's window-based approach limits the length of the text spans being considered at a time, making the capturing of long dependencies less feasible.

The ability of the neural approaches to output the relevant PN phrases that correspond to each feature greatly improves the interpretability of the scoring process by making explicit the relationship between the assigned score and its supporting evidence. Importantly, this is an improvement not only upon INCITE but also upon human scoring, as raters rarely have the time capacity to mark specific expressions. As each human rater scores hundreds of patient notes, it is not practically feasible for them to link specific phrases to rubric features for a large volume of data. In addition to improving interpretability, outputting the phrases enables applications of these tools that go beyond summative assessment. Such information can serve to provide pointed learner feedback in OSCE assessment, especially in cases where students are still learning how to document their clinical findings in an appropriately detailed and organized manner.

When discussing the development of this system, it is important to mention community competitions as an important source of innovation. The benefits

from sharing data for such purposes are not limited to the organization or the data science community, but extend to improving transparency – a crucial prerequisite for building stakeholder trust. When applying these creative approaches to a real-world scenario, important considerations such as speed and scalability limit the use of large model ensembles that are typically widely used in competitions. Other practical considerations include data availability for training (e.g., newly developed cases rarely have large numbers of PNs associated with them) and the need for weak supervision.

12 Limitations and ethical considerations

Some of the limitations of this research relate to the small within-case sample size of the annotated notes (which is somewhat mitigated by the large number of clinical cases) and the fact that not all notes could be double-rated due to resource constraints. While the scoring method is interpretable in that it can be traced to specific phrases within the PNs, the neural algorithms that identify the phrase boundaries are black-box models which needs to be carefully scrutinized for bias. In addition, it is still not fully apparent why certain features are easier to detect than others. Future work includes development of scoring approaches for other segments from the PNs such as the Physical Examination and Data Interpretation sections, deeper exploration of challenges related to specific features, experimentation with adversarial training, as well as further investigation of the operational use of the system.

Like many other products, automated scoring tools are socio-technical systems, whose impact is determined not solely by their technical capabilities but also by their use and output interpretation. Misuse and incorrect interpretation of the model outputs can lead to unethical practices of serious consequence. In a summative setting, the models described here are intended to be used as hybrid systems, where borderline cases and the notes from examinees below the passing standard are always reviewed by human raters. In a formative setting, it is paramount to carefully examine the relationship between use of the system and learning outcomes as necessary validity evidence.

Another ethical consideration for this study is the transparency of the approaches when developing technology for highly consequential decisions. As Spadafore and Monrad (2019) write: “decisioning software with the potential to profoundly affect the career of a medical student should be examined closely. Transparency of implementation is critical for such a high-stakes application". This is particularly important in automated scoring, where the scores only have value if all stakeholders (e.g., faculty, students, and residency selection programs, to name a few) trust that they are fair, reliable, and valid. Having public datasets and code such as the ones shared in the Kaggle competition go a long way in building trust by increasing transparency and accountability. As per the rules of the Kaggle competition[11], all code shared publicly is licensed under an Open Source Initiative-approved license. It is important to note that the benefits of system transparency go hand-in-hand with risks associated with using that knowledge to "game" the system. These include reverse-engineering a strategy that would result in a higher score, as well as the occurrence of negative “washback" (Green, 2013) – over-focus on developing only those skills that are currently covered by the scoring tool. Limiting these negative consequences while also building trust through transparency requires a delicate balance. In the case of this study, we foster transparency via organizing the competition, describing the main approaches, and evaluating our system on a dataset we made public. At the same time, we do not publish the code behind the system, limiting potential efforts reverse-engineer it or "game" it.

The data used in the Kaggle competition was released following strict adherence to ethical practice. It contains PNs only from examinees who explicitly indicated that they agreed to have their data used in research as part of the official exam registration process; Use of the anonymized data was considered “exempt" following an IRB review. The PNs were assigned a new set of IDs that cannot be linked to operational IDs used in scoring. None of the PNs include names, affiliations or personal descriptions (note that the names and clinical data associated with the standardized patients do not belong to real people; they are part of carefully constructed clinical cases that aim to resemble real-world clinical practice). In addition, the dataset does not feature complete PNs (only history portions are included), and no identifying information is given on which PNs were written by an individual examinee. According to Kaggle’s terms and conditions, data can only be accessed for partici-

[11] https://www.kaggle.com/competitions/nbme-score-clinical-patient-notes/rules

pating in the competition. For purposes unrelated to the competition, access to the data is subject to an application process and a data use agreement as a way to ensure ethical use.

A few important aspects remain to be examined before the system can be used in practice. This includes analyses related to differential functioning of the system for users with different backgrounds, e.g., ensuring that non-native English speakers are not disproportionally penalized due to differences in language proficiency, as well as continuous monitoring for issues such as drift or latency.

References

Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. 2020. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Barbara Barzansky and Sylvia I Etzel. 2016. Medical schools in the united states, 2015-2016. *JAMA*, 316(21):2283–2290.

Anthony Green. 2013. Washback in language assessment. *International Journal of English Studies*, 13(2):39–51.

Suchin Gururangan, Ana Marasovic, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. *CoRR*, abs/2004.10964.

Le Ha, Victoria Yaneva, Polina Harik, Ravi Pandian, Amy Morales, and Brian Clauser. 2020. Automated prediction of examinee proficiency from short-answer questions.

Polina Harik, Janet Mee, Christopher Runyon, and Brian E Clauser. 2023. Assessment of clinical skills: a case study in constructing an nlp-based scoring system for patient notes. In *Advancing Natural Language Processing in Educational Assessment*, pages 58–73. Routledge.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.

Beata Beigman Klebanov and Nitin Madnani. 2020. Automated evaluation of writing–50 years and counting. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 7796–7810.

Syed Latifi, Mark J Gierl, André-Philippe Boulais, and André F De Champlain. 2016. Using automated scoring to evaluate written responses in english and french on a high-stakes clinical competency examination. *Evaluation & the health professions*, 39(1):100–113.

Hieu Pham, Zihang Dai, Qizhe Xie, and Quoc V Le. 2021. Meta pseudo labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11557–11568.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*.

Abeed Sarker, Ari Z Klein, Janet Mee, Polina Harik, and Graciela Gonzalez-Hernandez. 2019. An interpretable natural language processing system for written medical examination assessment. *Journal of biomedical informatics*, 98:103268.

Maxwell Spadafore and Seetha U Monrad. 2019. Algorithmic bias and computer-assisted scoring of patient notes in the usmle step 2 clinical skills exam. *Academic Medicine*, 94(7):926.

King Yiu Suen, Victoria Yaneva, Janet Mee, Yiyun Zhou, Polina Harik, et al. 2023. Acta: Short-answer grading in high-stakes medical exams. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 443–447.

Victoria Yaneva, Janet Mee, Le An Ha, Polina Harik, Michael Jodoin, and Alex Mechaber. 2022. The usmle® step 2 clinical skills patient note corpus. Association for Computational Linguistics.

Wen-wai Yim, Ashley Mills, Harold Chun, Teresa Hashiguchi, Justin Yew, and Bryan Lu. 2019. Automatic rubric-based content grading for clinical notes. In *Proceedings of the Tenth International Workshop on Health Text Mining and Information Analysis (LOUHI 2019)*, pages 126–135.

Jianing Zhou, Vyom Nayan Thakkar, Rachel Yudkowsky, Suma Bhat, and William F Bond. 2022. Automatic patient note assessment without strong supervision. In *Proceedings of the 13th International Workshop on Health Text Mining and Information Analysis (LOUHI)*, pages 116–126.

A Example of a patient note

See Table 3 below.

B Annotation Guidelines

For each case, two of the notes were annotated jointly by a pair of annotators as part of an initial discussion to resolve discrepancies in the approach, with the next 5 notes annotated independently and discussed in a follow-up meeting. After that each annotator would proceed to independent work, where a subset of the notes were double-rated

History: Describe the history you just obtained from this patient. Include only information (pertinent positives and negatives) relevant to this patient's problem(s). Karin Moore is a 45 yo F here for nervousness. She recently noticed that she was feeling more nervous than usual and that this feeling has been progressively worsening. Nothing seems to help her nervousness. It is exacerbated by family and work. She feels especially nervous on Sunday night and Monday morning when as she is planning for the week. Unable to fall asleep and doesn't have appetite, though she does makes sure to eat. She denies significant changes in appetite, weight loss, or overall wellbeing. No fevers, chills, nausea, constipation, diarrhea, skin changes, racing heart, shortness of breath, dizziness, headaches or rashes. ROS: otherwise negative PMH: None; PSH: None Meds: Tylenol for occasional HA FHX: Father died at 65yo, had an MI Allergies: NKDA SH: Lives at home with husband, mother, and youngest son. Teaches literature at a local college. Has 2 drinks/mo, no tobacco or drug use.
Physical Examination: Describe any positive and negative findings relevant to this patient's problem(s). Be careful to include only those parts of examination you performed in this encounter. VS: Blood Pressure: 130/85 mm Hg Heart Rate: 96/min Gen: No acute distress, conversational, thin Neck: No thyromegaly, no lymphadeopathy Heart: RRR, no murmurs, rubs or gallops. Radial pulses +2 bilaterally Lungs: Clear to ascultation bilaterally, no wheezes Psych: Well-groomed. Non-pressured speech, linear though process.
Data Interpretation: Based on what you have learned from the history and physical examination, list up to 3 diagnoses that might explain this patient's complaint(s). (...) General anxiety disorder Panic disorder Hyperthyroidism

Table 3: Illustration of a PN. The dataset features only the history portions of the PNs.

for measuring agreement (10% for the proprietary data and 20% for the public data).

The annotators were given the following instruction:

- Identify all phrases that are expressions of a feature from the History portion of the PNs and link them to their corresponding feature.

- Include fragmented annotations by excluding the text that is not relevant to the feature (e.g., if the feature is *No relief with Imodium or Cipro*, only the underlined text of the following excerpt should be annotated: *Has tried Immodium (aggrevated condition), and Cipro 250mg BID (has taken 9 tablets) from prior episode of diarrhea in Kenya of lesser severity (no effect)*)

- Each feature should be marked up as a separate annotation, and the annotation should include all, but not more than, the text that captures the meaning of the corresponding entry in the feature (e.g., if the key essential is *No blood in stool*, only the underlined text of the following excerpt should be annotated: *No blood or mucus in stool*).

- Annotations should include quantifiers (e.g., *twice, four times, some*), intensifiers (e.g., *mild, severe*), and temporal modifiers (e.g., *two weeks, several years*) that are specified in the corresponding entry in the feature, as well as the object that is being described (e.g., *pain, cough*).

- Annotations should not include articles (e.g., *a, the*) or references to the patient (e.g., *her, he*) that occur at the beginning of note entries, or end punctuation (e.g., periods); however, it is not necessary to fragment annotations if words or characters, such as these, occur within relevant text and do not modify the meaning of the feature entry.

- Annotations may overlap; that is, they may share text with other annotations. For example, negations (e.g., *negative for, no, denies*) frequently will be shared among several annotations. In the phrase *Negative for fever, chills, nausea, vomiting, hematochezia*, the negated

nouns refer to different features and should be annotated as Negative for fever, *Negative for chills, Negative for nausea*, etc.

- Mark up every instance of the feature whether it is identical to an existing annotation or not. For example, if the feature is *NSAID-use* and the examinee wrote *Uses NSAIDs* as well as *took ibuprofen*, both snippets of text should be annotated. If the exact snippet *Uses NSAIDs* appeared more than once in a note, it should be annotated every time it appears in the note.

- Gender is a special case of a feature and should only be annotated once for the first mention. Subsequent phrases that may be linked to gender such as *she* or *his* should not be annotated.

C AWS System Architecture

See Figure 3 below for a visualization of the system architecture.

D Extended Error Analysis

Examples of features that were always correctly identified include *'no previous uti', 'occasional morning headaches', 'no temperature intolerance or no weight change or no bowel changes or no hair changes or no skin changes', 'on depo provera',* and *'decreased energy or fatigue'*. The top 5 features with most FPs were *getting worse* (7), *hand stiffness* (5), *subjective fever* (5), *chest pain with cough* (5), and *overdue for colonoscopy* (5). The top 5 features that were most difficult to detect automatically with highest numbers of FNs were *1 day urinary frequency* (4), *radiating down back of neck* (3), *constipation x 4 5 months* (3), *acute onset* (3), *nausea* (3). There was no apparent pattern as to what made certain features easy or challenging to detect, with both groups containing negation, temporal aspects, and features with varying length in characters.

The case with the highest number of errors (n = 18) contained 31 features to look for. Out of the 18 errors, 10 were FPs, and out of these, 4 features looked for negated terms (*no change in diet, no oral contraceptives, no abdominal surgeries* and *no radiation*). Interestingly, some negated expressions from the PNs were erroneously mapped to these negated features such as *denies eating under cokked [sic] foods* being mapped to *no change in diet*, showing that the model is aware that it needs to look for negation but processing it incorrectly. The remaining eight FNs did not reveal a pattern.

Of all errors, 594 were false positives (FPs) across 166 cases and 396 were false negatives (FNs) across 151 cases. The highest number of FPs per case was 12 (2 cases), with the majority of cases containing one or two FPs per case (34 and 35 cases, respectively). For FNs, the highest number of FNs per case was 9 (1 case), with the majority of cases also containing one or two FNs (48 and 37 cases, respectively).

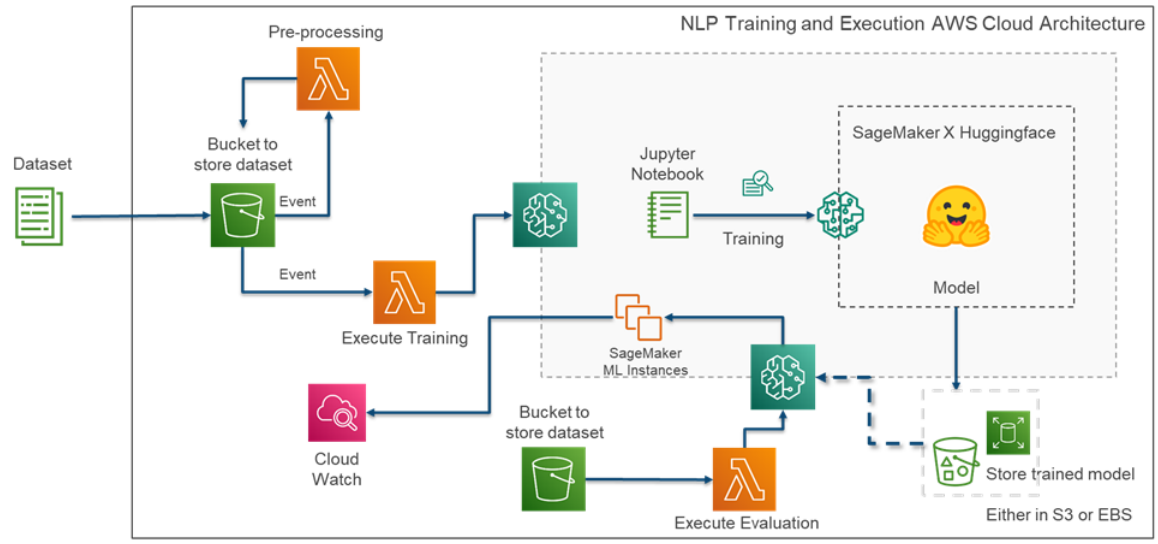

Figure 3: AWS System Architecture. When a new dataset is placed in the S3 bucket, a Lambda function triggers preprocessing and a subsequent Lambda function triggers the training process. Training is performed via SageMaker and Huggingface; final predictions are stored in CloudWatch.