Automatic rubric-based content grading for clinical notes

Wen-wai Yim, Harold Chun, Teresa Hashiguchi, Justin Yew, Bryan Lu
Augmedix, Inc.
{wenwai.yim,haroldchun,teresa, justinyew,bryan}
@augmedix.com

Ashley Mills
University of Cincinnati
College of Medicine
mills2an@mail.uc.edu

Abstract

Clinical notes provide important documentation critical to medical care, as well as billing and legal needs. Too little information degrades quality of care; too much information impedes care. Training for clinical note documentation is highly variable, depending on institutions and programs. In this work, we introduce the problem of automatic evaluation of note creation through rubric-based content grading, which has the potential for accelerating and regularizing clinical note documentation training. To this end, we describe our corpus creation methods as well as provide simple feature-based and neural network baseline systems. We further provide tagset and scaling experiments to inform readers of plausible expected performances. Our baselines show promising results with content point accuracy and kappa values at 0.86 and 0.71 on the test set.

1 Introduction

Clinical notes, essential aspects of clinical care, document the principal findings of the visit, hospital stay or treatment episode, including complaints, symptoms, relevant medical history, tests performed, assessments and plans. During an encounter or soon after, notes are created based on subjective history, objective observations, as well as clinician assessment and care plans. Although this is a regular aspect of clinical care in all institutions, there is a large variability in the details taken. Clinical documentation training is often informal and institution-dependent, as systematic training of clinical documentation can be time-consuming and expensive. Training involves continued monitoring of note quality and completeness.

In this work, we present the problem of clinical note grading and provide several baseline systems for automatic content grading of a clinical note given a predefined rubric. To solve the problem, we built a simple feature-based system and a simple BERT-based system. We additionally provide training size experiments and tagset experiments for our baseline systems, as well as experiment with the relevance of using Unified Medical Language System (UMLS) and similarity features.

2 Background

Clinical notes serve as critical documentation for several purposes: medical, billing, and legal requirements. At the same time, a clinical note is written, updated, and consumed for the purpose of communication between clinicians for clinical care over a period of time. Too much irrelevant information can become an impediment to clinical care. Therefore, the identification of a base level of information is required to assess a note.

At our institution, clinical note documentation (or medical scribe) trainees are assessed via quizzes and exams in which trainees produce clinical notes based on mock patients visits (written to mimic outpatient encounters). Clinical note responses are graded against a grading rubric by training specialists, who have medical scribing experience as well as specialized training in determining note quality. The goal is to train scribes to produce a clinical note based on listening to the patient-clinician conversation during a clinical visit. Thus, the scribe expected to actively producing content, not just merely transcribe dictations from a clinician.

The purpose of a note rubric is to encapsulate the base requirements of a clinical note. Rubrics contain 40-60 rubrics items which reflect the information that needs to be captured during a medical encounter. A rubric item is typically written as a phrase and includes medically relevant attributes. For example, a rubric item discussing a

Proceedings of the 10th International Workshop on Health Text Mining and Information Analysis (LOUHI 2019), pages 126–135
Hong Kong, November 3, 2019.
https://doi.org/10.18653/v1/D19-62

symptom will typically require information about duration and severity. A rubric item discussing medication will often include dosage information. Each rubric is associated with a section of the note where it needs to be placed. For training purposes, standard note sections include: History of Present Illness (HPI), a detailed subjective record of the patient's current complaints; Review of Systems (ROS), the patient's subjective complaints grouped by organ system; Physical Exam (PE), the clinician's objective findings grouped by organ system; and Assessment and Plan (AP), the clinician's diagnosis and the next steps in treatment for each diagnosis. Figure 1 gives an example of a clinical note with these sections.

If the note contains text that satisfies a rubric item, then a content point for that rubric item is awarded. If the note contains an incorrect statement (e.g. the wrong medication dosage), then that rubric point is not awarded, regardless of a correct statement appearing elsewhere. If the note lacks the inclusion of a rubric point, then that rubric point is not awarded. At most, one content point can be awarded per rubric item. Examples of several rubric items with corresponding portions of a clinical note are shown in below.

Rubric item examples

- frequent_bm_3-4_times_per_day (HPI section), documents relevant symptom history – *"The patient complains of frequent bowel movements, 3-4 times daily."*
- pe_skin_intact_no_clubbing_cyanosis (PE section), documents physical exam performed during visit – *"Skin: Skin intact. No clubbing or cyanosis. Warm and dry."*
- plan_advise_brat_diet (AP section), documents that the provider recommended the BRAT diet to the patient – *"Recommended that the patient follow the BRAT diet for the next few days. "*

CHIEF COMPLAINT: Frequent urination

HISTORY OF PRESENT ILLNESS:
The patient is a 33 year old female who presents today complaining of frequent urination and bowel movements...
...
REVIEW OF SYSTEMS:
Constitutional: Negative for fevers, chills, sweats.
...
PHYSICAL EXAM:
General: Temperature normal. Well appearing and no acute distress
...
ASSESSMENT & PLAN:
1. Ordered urinalysis to rule out urinary tract infection
2. Put her on brat diet, counseled patient that BRAT diet is...
...

Figure 1: Abbreviated example of a clinical note. Clinical notes are typically organized by sections. The exact section types and ordering in real practice my vary by specialty and organization.

3 Related Work

Most community efforts in automatic grading have been in the context of automatic essay grading (AEG) and automatic short answer grading (ASAG), both of which harbor significant differences than our rubric-based content grading task.

AEG involves rating the quality of an essay in terms of coherence, diction, and grammar variations. Typically, an essay is given a score, e.g. from 1-6. Applications include essay grading for standardized tests such as for the GMAT (Graduate Management Admission Test) or TOEFL (Test of English as Foreign Language) (Valenti et al., 2003). Key architects for these systems are often commercial organizations. Examples of commercial computer-assisted scoring (CAS) systems include E-rater and Intelligent Essay Assessor. Systems such as E-rater use a variety of lingustic features, including grammar, diction, and as well as including discourse level features (Attali and Burstein, 2006; Zesch et al., 2015). In another approach, the Intelligent Essay Assessor uses latent semantic analysis to abstract text to lower-rank dimension-cutting representations of documents. Scores are assigned based on similarity of new text to be graded to a corpus of previously graded text (Foltz et al., 1999). The release of the Kaggle dataset has made this type of data more available to the public (kaggle.com/c/asap aes). A key difference of AEG task from our grading task is that our efforts focus on specific content item grading and feedback, over a single holistic document level rating.

In ASAG, free text answers to a prompt are graded categorically or numerically. It is very closely related to paraphrasing, semantic similarity, and textual entailment. System reporting for this task has often been on isolated datasets with a wide range of topics and setups. Often, these systems require extensive feature engineering (Burrows et al., 2015). One example system is C-rater, produced by ETS, which grades based on the presence or absence of required content (Lea-

cock and Chodorow, 2003; Sukkarieh and Blackmore, 2009). Each required piece of content, similar to our rubric, in the text is marked as absent, present, negated, with a default of not_scored. Text goes through several processing steps, including spelling correction, concept recognition, pronoun resolution, and parsing. These features are then sent through a maximum entropy model for classification. Semantic similarity approaches apply a mixture of deep processing features, e.g. shortest path between concepts or concept similarities (Mohler and Mihalcea, 2009). In the SemEval 2013 Task 7 Challenge, the task involved classification of student answers to questions, given a reference answer, and student answers. Student answers are judged to be correct, partially_correct_incomplete, contradictory, irrelevant, or non_domain (Dzikovska et al., 2013). Although it has much in common to our rubric-based content grading setup, short answer grading has less document level issues to contend with. Moreover, specifically for our case, document-level scoring has some non-linearity with the individual classification of sub-document level text, e.g. finding one contradictory piece of evidence negates a finding of a positive piece of evidence.

The work of (Nehm et al., 2012), which attempts to award content points for specific items for college biology evolution essays, most closely resembles our task. In the task, students are awarded points based on whether they articulate key natural selection concepts, e.g. familiar plant/trait gain (mutation causing snail to produce poisonous toxin would increase fitness). The authors experimented with configuring two text analytic platforms for this task: SPSS Text Analysis 3.0 (SPSSTA) and Summarization Integrated Development Environment (SIDE). SPSSTA requires building hand-crafted vocabulary and attached rules. SIDE uses a bag-of-words representation run through a support vector machine algorithm for text classification. Key differences from our task are that rubric items are more numerous and specific; furthermore, our medium is a clinical note, which has documented linguistic and document style differences than normal essays; finally, our goal is not only to grade but to give automated in-text feedback.

In this work, we present our system for grading a clinical note given a grading rubric, which also gives subdocument level feedback. To our knowledge, there has been no previous attempt at clinical note automatic content grading.

4 Corpus Creation

We created a corpus by grading training notes by multiple different trainees quizzed on the same mock patient visit quiz, which included 40 rubric items. Annotation was carried out using using the ehost annotation tool (South et al., 2012). Annotators were asked to highlight sentences for if they were relevant to a rubric item. Furthermore, they were to mark whether a highlight had one of four attributes: correct, incorrect_contrary, incorrect_missingitem, and incorrect_section.

frequent_bm_3-4_times_per_day attribute examples

- correct –*"The patient reports having 3-4 bowel movements a day."* (Located in the HPI)
- incorrect_contrary –*"The patient has been having bowel movements every 30 minutes."* (Located in the HPI) Explanation: The frequency is much higher than what would be expected for 3-4 times per day. Thus the information content is considered to be inaccurate or contrary to what is given by the rubric.
- incorrect_missingitem – *"The patient reports having many bowel movements each day."* (Located in the HPI) Explanation: This statement does not give any inaccurate information but is missing a crucial element that is required to earn this rubric point, which is the frequency value is 3-4 times per day.
- incorrect_section – *"The patient reports having 3-4 bowel movements a day."* (Located in the AP) Explanation: This statement is correct, but is located in the wrong section of the note.

Parts of the note were marked for exclusion, such as short hand text (excluded because they are just notes taken by the scribe in training of the conversation) and the review of systems (ROS) part of the note (this was excluded because grading of that section was not enforced at its time of creation). Entire notes were marked for exclusion in cases where the note was intended for a different exam or in cases when the note contained all short hand and templated sections (e.g. only notes and section headers such as "Chief Complaint"). Discontinuous rubric item note contexts were linked together with a relation. The final corpus after included 338

	A1	A2	A3	A4	A5
A1	-	0.84/0.68	0.84/0.69	0.86/0.72	0.85/0.69
A2	-	-	0.86/0.72	0.87/0.73	0.87/0.73
A3	-	-	-	0.87/0.73	0.85/0.69
A4	-	-	-	-	0.88/0.76

Table 1: Content-point inter-annotator agreement (percent identity/kappa).

	A1	A2	A3	A4	A5
A1	-	0.84/0.68	0.84/0.69	0.86/0.72	0.85/0.69
A2	-	-	0.86/0.72	0.87/0.73	0.87/0.73
A3	-	-	-	0.87/0.74	0.85/0.69
A4	-	-	-	-	0.88/0.76

Table 2: Offset-level inter-annotator agreement (label/label-attribute).

notes, with a total of 10406 highlights[1]. A total of 244 rubric items required connecting discontinous highlights. The full corpus statistics are shown in Table 3.

Inter-annotator agreement was measured among 5 annotators for 9 files. We measured agreement at several levels. At a note level, we measured pairwise percent identity and Cohen's kappa (McHugh, 2012) by content points. At the text offset level, we measured precision, recall, and f1 (Hripcsak and Rothschild, 2005) for inexact overlap of highlights both at the label level (e.g. cc_frequent_urination) and at a label attribute level (e.g. cc_frequent_urination:correct). Since incorrect_section is not counted in content-based grading, for both inter-annotator agreement and for subsequent analysis, incorrect_section highlights were counted as correct unless overlapping with a incorrect_missingitem highlight, which would make it count as incorrect_missingitem. The agreement scores are shown in Table 1 and 2. Fleiss kappa (Fleiss, 1971) at the content point level was at 0.714. The rest of the corpus was divided among 5 annotators.

5 Methods

We performed classification for two types of systems: a feature-based and a simple BERT based neural network system for text classification. Since discontinuous text highlights accounted for less than 10% of items, we choose not to model this nuance. Both systems used the same pre-processing pipeline configurations shown in Figure 2.

[1]Includes highlights for excluding the note, as well as excluding short hand text and ROS sections

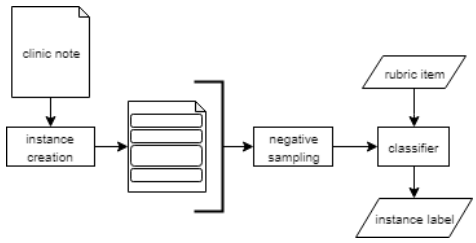

Figure 2: General pipeline

We split 338 files into 270 training, 68 test set. Tuning was performed on the training set in 5-fold cross validation.

5.1 Pipeline configurations

The text was preprocessed, where short-hand text and blank lines were removed. Sentences and words were tokenized using spaCy (spacy.io).

Instances were created by extracting minichunks (sentences or subsections in the case of the PE section) from the clinical note identified using regular expressions. We experimented with three tag-set configurations (tag_gran) with values (2lab, 3lab, 4lab), which represents {1 vs 0 points}, {correct, incorrect_contrary, vs missing} and {correct, incorrect_contrary, incorrect_missingitem, missing} per rubric item. Missing is the default value if no relevant highlight annotates the sentence.

For the minichunk level, we set a configuration negative sampling ratio (neg_samp) which specifies the factor of default class instances to non-default class instances. A similarity feature flag (simfeat_on) turns on similarity features in the feature-based system or switch the BERT-based system to one which includes matching features. For our feature based modeling, we used scikit-learn (Pedregosa et al., 2011), for our neural network pipelines we incorporated allennlp (Gardner et al.).

5.2 Feature based system

The feature based system includes an n-gram feature extraction, which then passed through a chi-squared statistical feature selection, before using a support vector machine implemented by scikit learn svc.

If the umls configuration is turned on, then Unified Medical Language System (UMLS) concept with its negation value, extracted using MetaMap, concept-grams are also added (Aronson and Lang, 2010). If turned on, similarity features for n-grams

rubric_item	total	correct	incorrect_missingitem	incorrect_contrary	incorrect_section
cc_frequent_urination	579	535	9	35	0
checks_blood_sugar_regularly_110s_before_meals	152	39	86	26	1
bp_fluctuates_150100_	365	239	91	34	1
denies_abdominal_pain_vomiting_constipation	270	98	169	3	0
denies_hematochezia	211	202	1	8	0
denies_recent_travel	257	245	5	4	3
duration_of_1_week	256	221	5	30	0
feels_loopy	125	110	8	6	1
feral_cat_in_the_house_occasionally	175	56	95	22	2
frequent_bm_3-4_times_per_day	343	249	50	44	0
frequent_urination_every_30_minutes	347	276	31	39	1
has_not_had_ua	190	177	7	6	0
healthy_diet	178	157	21	0	0
husband_was_sick_with_uti_symptoms	331	257	56	18	0
hx_of_htn	298	254	40	4	0
initially_thought_she_had_respiratory_infection	204	78	33	93	0
loose_stools_with_mucous	324	205	111	8	0
losartan_hctz_every_night_with_dinner	325	148	154	23	0
mild_dysuria	266	185	57	24	0
no_recent_antibiotics	279	263	10	5	1
pe_abdomen_hyperactive_bowel_sounds_at_llq_no_pain_with_palp	334	97	180	51	6
pe_cv_normal	315	300	10	4	1
pe_extremities_no_edema	297	282	7	2	6
pe_heent_normal_no_thyromegaly_masses_carotid_bruit	430	173	251	4	2
pe_resp_normal	331	324	6	1	0
pe_skin_intact_no_clubbing_cyanosis	276	84	173	6	13
plan_advise_brat_diet	312	249	42	15	6
plan_bp_goal__13080	155	123	11	18	3
plan_may_notice_leg_swelling_notify_if_unbearable	216	93	114	5	4
plan_prescribed_amlodipine_5mg	268	205	40	18	5
plan_recommend_30_mins_physical_activity_4-5_times_per_week	296	128	131	34	3
plan_reduce_stress_levels	119	100	13	1	5
plan_rtn_in_1_month_with_continued_bp_log	280	172	85	16	7
plan_ua_today_to_rule_out_infx	302	202	91	3	6
side_effects_of_difficulty_breathing_with_metoprolol	164	104	38	21	1
stress_work_related	223	215	7	1	0
takes_blood_pressure_every_morning	222	145	66	11	0
tried_yogurt_and_turmeric_no_improvement	176	66	106	4	0
was_seen_by_dr_reynolds_yesterday	154	44	96	13	1
weight_normal	61	52	5	3	1

Table 3: Label frequencies for full corpus

and umls concept grams (if the umls flag is on) are also added. We used jaccard similarity between 1-, 2-, and 3- grams. The full configurations include the following :

- top_n : top number of significant features to keep according to chi-squared statistic feature selection (integer)
- sec_feat : setting to determine how section information is encoded. If "embed" is set, then each feature with be concatenated with its section, e.g. "sect[hpi]=patient". If "sep" is turned on, "sect=hpi" is added as a feature.
- umls : whether or not to use umls features (binary)
- simfeat_on : whether or not to turn on similarity features (binary)
- text_off : whether or not to turn off the features not related to similarity
- umls_text_off : whether or not to turn off the umls features not related to similarity
- sent_win : window for which surrounding sentence features should be added, e.g. sent[-1]=the would be added as a feature from previous sentence unigram feature "the" if sent_win=1. (integer)

5.3 BERT based system

The neural network system made use of the previous instance creation pipeline; however in place of feature extraction, instances were transformed into BERT word vector representations. We used the output for CLS position of the embeddings to represent the whole sequence similar to that of the original paper (Devlin et al., 2018).

To mimic the feature-based system's case of simfeats_on, we include a switch to an architecture that also feeds in the CLS position output from a paired BERT classification setup. When simfeats_on is turned off, the architecture becomes that of a simple BERT sentence classification (bert). When text_off is turned on, then the architecture becomes that of a simple BERT sentence pair classification (bertpair). When simfeats_on is turned on and text_off is turned off, we have a system with both types off representations (bert+bertpair). A figure of the BERT classifier setup is show in Figure 3.

Because certain medical vocabulary may not be available with the general English trained cor-

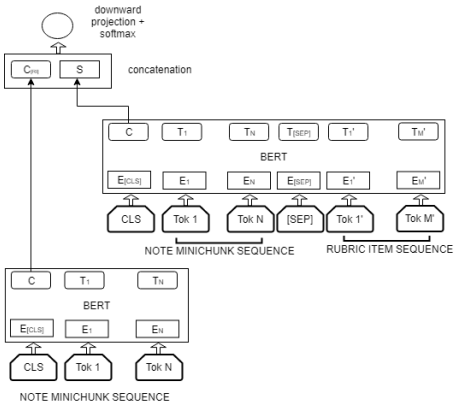

Figure 3: BERT with additional sentence pair classifier

pora, we used pre-trained BERT embedding and vocabulary from bio-bert, which is fine-tuned on pubmed data (Lee et al., 2019).

5.4 Simple baseline

For a further comparison, we have included a feature-based document-based baseline. This baseline largely follows the previously mentioned feature-based baseline though performed at the document level. Because it is performed at a document level, some attribution to a sub-document level unit becomes necessary. (Recall, we wish to be able to identify which part of the document is relevant to a rubric item as well as if we believe it is correct or otherwise.) To identify a corresponding offset labeling for this setting, we attributed a document level classification to a sentence which contained the maximum amount of important features. We defined features to be important according to their learned feature weight magnitude crossing a configurable threshold value. Thus, for a document classification, based on this logic we are able to assign a sentence related to that classification, for which we can use all our previously mentioned metrics for evaluation. We found a threshold of 10 to work well in our experiments.

5.5 Evaluation

Similar to inter-annotator agreement, we measured performance using several metrics. At the note level, we measured distance to target full document score by mean absolute error (MAE). We also measured content point accuracy and kappa to get a sense of the performance in point assignment.

At the offset level, we measured precision, recall, and f1 for rubric item label-attribute value. For minichunk classification, offsets were translated according to the start and end of the minichunk in the document.

6 Results

Evaluations in cross-validation are reported as the average of all 5 folds. Consistent with this, graphical error bars in the learning curve figures represent the standard deviation across folds.

6.1 Experimental configurations

Feature-based parameter tuning. We started with tag_gran at 4lab, simfeats_on true, and top_n set at 1000, sent_win=0, and neg_samp=2. We then greedily searched for an optimal solution varying one parameter at a time to optimize precision, recall and f1 measure. Our final configurations were set to neg_samp=10, sect_feat set to "sep", top_n=4000, sent_win=0. We kept the same configurations for the other feature-based systems.

Neural network hyperparmeter tuning.
For the neural network models, we mainly experimented with negative sampling size, dropout, and context length. We found a neg_samp=100, dropout=0.5, epochs=2 and context_len=300 to work well.

6.2 Cross-validation and test results

Table 4 shows performances for the feature-based system using different tagsets in cross-validation. Unsurprisingly the most detailed label (4lab) at the more granualar level (minichunk) showed the best performance. Tables 4 and 5 shows the comparison of different system configurations in evaluated in cross validation. Table 6 shows results for the test set for different systems. In general, the feature-based system with the full feature-set out performed for the cross-validation and test experiments. Among the BERT systems, the simple BERT system did better than the other two configurations.

6.3 Scaling and tagset experiments

Figure 4 shows the effect of increasing training size for several metrics, under several select systems. Interestingly, 2lab and 3lab settings show similar behaviors across different metrics. For the document level baseline, tagset does not make any large difference across all three metrics. Different from other systems, 2lab and 3lab setting,

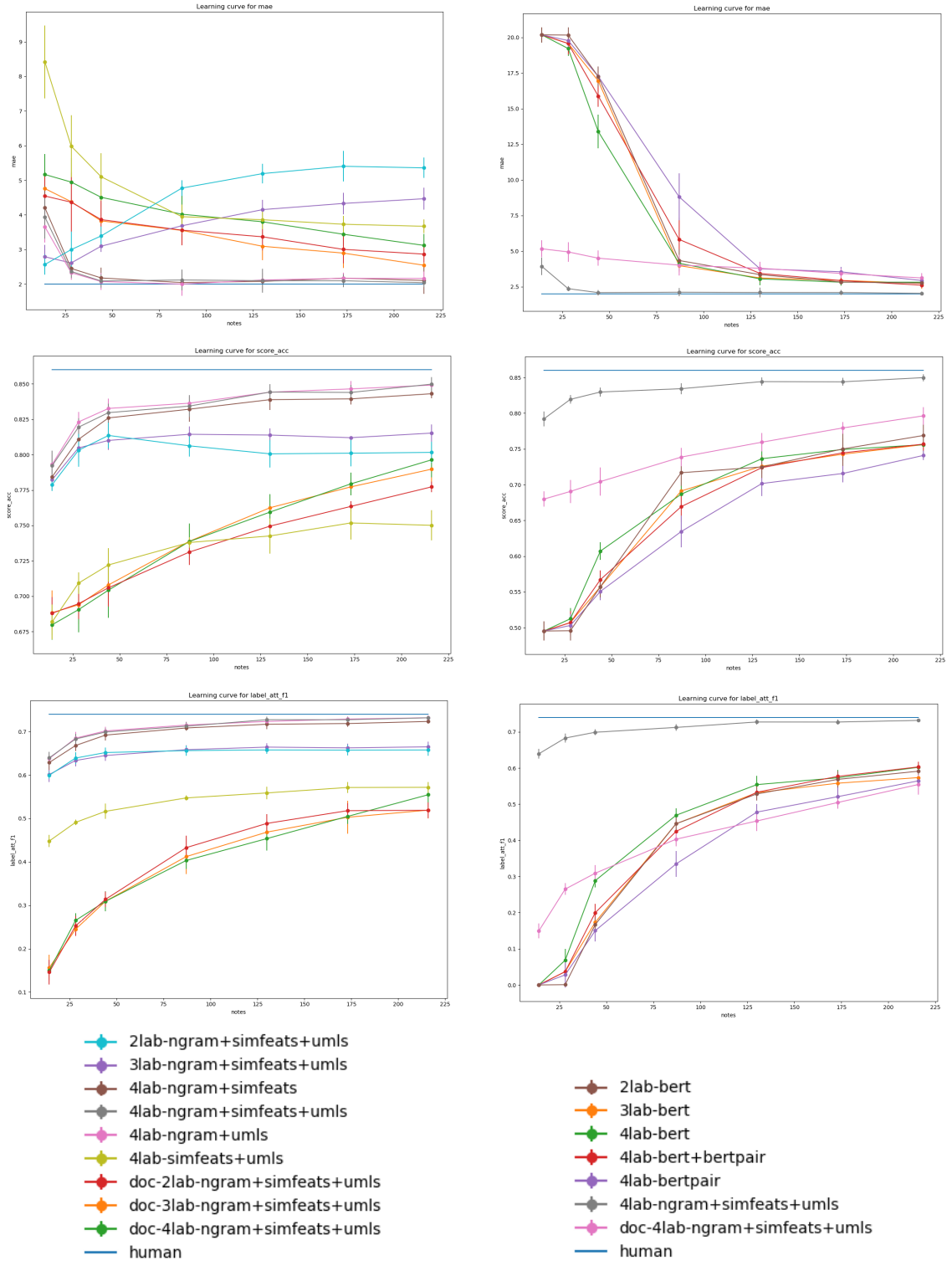

Figure 4: Learning curve experiments. 2lab, 3lab, 4lab demarks the tagset configuration. doc-* identifies the simple document classification baseline. The left column shows the performance of different n-gram configurations for 3 performance metrics. The right column shows BERT system performances for the 3 metrics along with two feature-based systems for comparison.

tagset	mae	acc	kappa	p	r	f
4lab	2.0	0.85	0.70	0.71	0.75	0.73
3lab	4.5	0.82	0.63	0.69	0.64	0.66
2lab	5.4	0.80	0.60	0.69	0.63	0.66
doc-4lab	3.1	0.80	0.59	0.58	0.53	0.55
doc-3lab	2.5	0.80	0.58	0.65	0.43	0.52
doc-2lab	2.9	0.78	0.55	0.67	0.42	0.52

Table 4: Feature-based system results for 5-fold cross-validation, varying tagsets.

system	mae	acc	kappa	p	r	f
ngram+simfeats+umls	2.0	0.85	0.70	0.71	0.75	0.73
ngram+umls	2.2	0.85	0.70	0.71	0.75	0.73
ngram+simfeats	2.1	0.84	0.69	0.70	0.74	0.72
simfeats+umls	3.7	0.75	0.50	0.63	0.53	0.57
bert+bert_pair	2.6	0.76	0.51	0.59	0.62	0.60
bert	2.8	0.76	0.51	0.60	0.61	0.60
bert_pair	2.9	0.74	0.48	0.57	0.56	0.56

Table 5: Results for 5-fold cross-validation, tag_gran=4lab.

when trained to maximize both precision and recall equally for label attributes, MAE will rise instead of lower such as for the other system setups. This makes sense, as both tagset settings miss crucial examples that exhibit confusing features. For example, on the 2lab setting, only positive examples are shown not those that have incorrect information or those that have missing information (i.e. partially correct information). Likewise, the 3lab setting does not have evidence for partially correct items. We found experimentally, when tuning for higher precision and lower recall, that MAE also tends to lower– suggesting that these two settings can be better maximized by tuning for MAE instead.

Like the document-based baseline, the BERT systems' performances showed that tagset did not make as big of a difference across all three metrics at different training size levels. This is possibly because there were not enough examples to even properly fine-tune for these two systems which require more training data. At higher levels of training sizes, tagsets may again come into effect. Though the BERT systems at lower training sizes start at lower performances, it quickly catches up to the document classification baseline for the MAE and f1 metrics, though never gets close to the 4lab feature-based baseline.

system	mae	acc	kappa	p	r	f
ngram+simfeats+umls	2.5	0.85	0.70	0.68	0.78	0.72
ngram+umls	2.5	0.85	0.70	0.68	0.78	0.72
ngram+simfeats	2.3	0.86	0.71	0.68	0.79	0.73
simfeats+umls	3.8	0.77	0.54	0.63	0.55	0.59
bert+bert_pair	3.2	0.77	0.53	0.57	0.60	0.58
bert	2.8	0.79	0.57	0.59	0.63	0.61
bert_pair	3.1	0.76	0.51	0.58	0.60	0.59

Table 6: Detailed results for the test set, tag_gran=4lab.

6.4 UMLS and similarity features

The addition of similarity features did not provide a significant boost for the ngram feature-based system. Similarity features alone for the feature-based system underperformed at all training size levels compared to the ngram models. On the other hand, the addition of UMLS features increased performance across three metrics for all training size levels.

The BERT based system using only the simple BERT representation (without paired features), outperformed the other two settings across the three metrics at most training size levels in cross-validation. However, near higher levels of training data, BERT with BERT pair becomes comparable. The BERT pair system underperforms across all three metrics and at all training sizes.

6.5 Error Analysis

One challenging aspect of the classification task was the imbalanced categories across notes for different rubric items. Some labels were inherently less frequent, e.g. weight_normal had a total of 61 compared to cc_frequent_urination with 579 highlights. Indeed the performance amongst all rubric item scores was highly variable, with 13% f1 standard variation for the label-attribute measure. Moreover, the distribution of classes per label was also highly variable, as shown in Figure 3. For example, when no_recent_antibiotics or stress_work_related appears, in labeled data they are often correct. As a consequence, accurately predicting less populated classes becomes more difficult. For the best performing system, for example, there were instances where *"Patients weight is not normal"* was considered correct despite the rubric specifying the opposite. Similarly, *"Patient denies feeling loopy"* would be marked correct when the rubric says otherwise. When measuring at the label level for highlights instead, the performance on the test was higher by more than 10% f1, as shown in Table 7. This

eval	tp	fp	fn	p	r	f
label-att	1470	700	424	0.68	0.78	0.72
label	1708	413	186	0.81	0.90	0.85

Table 7: Results for the test set, ngram+simfeats+umls tag_gran=4lab.

indicates that many errors are due to confusion between classification categories. Contradictions, labeled incorrect_contrary, for this reason was a large problem.

Manually studying errors in the test set for the best performing system, we found that rubric items frequently identified in the training, were broadly correctly classified. However, there were some rubric items that had more inconsistencies in how they were being tagged or graded. Some errors were partly due to human grading error. For example, checks_blood_sugar_regularly_110s_before_meals was a rubric item that scribes frequently missed when creating notes. Due to this, some sentences with just *"checks"* were sometimes labeled for checks_blood_sugar_regularly_110s_before_meals despite the fact that the sentences were about checking blood pressure. This leads to cases where synonymous phrases to *"blood sugar"*, *"blood glucose"*, did not get labeled as instances with *"blood glucose"* by the human graders.

7 Conclusions

In this paper we present the problem of clinical note grading and provide several baseline system evaluations at different levels of training data. We show experimentally that the choice of labeling has large effects upon the system performance. Furthermore, though neural network systems may relieve a lot of feature-engineering, this may not be plausible for smaller corpora.

Further improvements can be made by rubric-item specific pipeline specialization, as well as further augmentation of specific feature extraction modules, e.g. better negation handling. Deeper processing methods and features, including use of lemma-ization and dependency structures, would make features more generalizable. On the other hand, to maximize performance, feature-extraction can also be made more rubric-item specific, for example by hand-crafting features. For this work, we used a linear support vector machine for our classifier, but further experimentation with different classifiers for each rubric item would lead to higher performances. The BERT systems can be improved by increasing training size and adding more feed-forward layers.

Our proposed system can be used to expedite and formalize clinical note creation in a training setting. For example, instead of having a human grader view all training notes, a simple pass of the automated grading system can eliminate those that will fail with some confidence. For others, a human grader can correct the output of the system, which would speed the grading process then if the grader had to mark highlights alone. In this work, we focus on cases for which we have many examples of clinical notes generated for the same encounter with a fixed rubric. Future work will investigate grading for arbitrary notes and rubrics.

Acknowledgments

We would like to thank the Augmedix training specialist team as well as all those involved with creating the source dataset for this work.

Very special thanks to Kevin Holub and Sonny Siddhu for their efforts in initiating Augmedix's efforts in AI-assisted in-text grading for which is the motivation behind this project.

References

kaggle.com/c/asap aes. The hewlett foundation: Automated essay scoring | kaggle.

Alan R Aronson and Franois-Michel Lang. 2010. An overview of MetaMap: historical perspective and recent advances. 17(3):229–236.

Yigal Attali and Jill Burstein. 2006. Automated essay scoring with e-rater v.2 | the journal of technology, learning and assessment. 4(3).

Steven Burrows, Iryna Gurevych, and Benno Stein. 2015. The eras and trends of automatic short answer grading. 25(1):60–117.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. abs/1810.04805.

Myroslava Dzikovska, Rodney Nielsen, Chris Brew, Claudia Leacock, Danilo Giampiccolo, Luisa Bentivogli, Peter Clark, Ido Dagan, and Hoa Trang Dang. 2013. SemEval-2013 task 7: The joint student response analysis and 8th recognizing textual entailment challenge. In *Second Joint Conference on Lexical and Computational Semantics (*SEM),*

Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), pages 263–274, Atlanta, Georgia, USA. Association for Computational Linguistics.

JL Fleiss. 1971. Measuring nominal scale agreement among many raters. 76(5):378–382.

Peter W Foltz, Darrell Laham, and Thomas K Landauer. 1999. IMEJ article - the intelligent essay assessor: Applications to educational technology.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew E. Peters, Michael Schmitz, and Luke S. Zettlemoyer. AllenNLP: A deep semantic natural language processing platform. abs/1803.07640.

George Hripcsak and Adam S Rothschild. 2005. Agreement, the f-measure, and reliability in information retrieval. 12(3):296–298.

Claudia Leacock and Martin Chodorow. 2003. C-rater: Automated scoring of short-answer questions. 37(4):389–405.

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.

Mary L McHugh. 2012. Interrater reliability: the kappa statistic. 22(3):276–282.

Michael Mohler and Rada Mihalcea. 2009. Text-to-text semantic similarity for automatic short answer grading. In *EACL*.

Ross H. Nehm, Minsu Ha, and Elijah Mayfield. 2012. Transforming biology assessment with machine learning: Automated scoring of written evolutionary explanations. 21(1):183–196.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Brett R. South, Shuying Shen, Jianwei Leng, Tyler B. Forbush, Scott L. DuVall, and Wendy W. Chapman. 2012. A prototype tool set to support machine-assisted annotation. In *Proceedings of the 2012 Workshop on Biomedical Natural Language Processing*, BioNLP '12, pages 130–139, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jana Z Sukkarieh and John Blackmore. 2009. c-rater: Automatic content scoring for short constructed responses.

Salvatore Valenti, Francesca Neri, and Alessandro Cucchiarelli. 2003. An overview of current research on automated essay grading. 2:319–330.

Torsten Zesch, Michael Wojatzki, and Dirk Scholten-Akoun. 2015. Task-independent features for automated essay grading. In *BEA@NAACL-HLT*.