REFLECTSUMM: A Benchmark for Course Reflection Summarization

Yang Zhong†*, Mohamed Elaraby†*, Diane Litman†,
Ahmed Ashraf Butt♣, Muhsin Menekse◇
† Department of Computer Science, School of Computing and Information
University of Pittsburgh, Pittsburgh, USA
♣ School of Computer Science, Carnegie Mellon University, Pittsburgh, USA
◇ School of Engineering Education, Purdue University, West Lafayette, USA
{yaz118, mse30, dlitman}@pitt.edu
ahmedasb@cs.cmu.edu
menekse@purdue.edu

Abstract

This paper introduces REFLECTSUMM, a novel summarization dataset specifically designed for summarizing students' reflective writing. The goal of REFLECTSUMM is to facilitate developing and evaluating novel summarization techniques tailored to real-world scenarios with little training data, with potential implications in the opinion summarization domain in general and the educational domain in particular. The dataset encompasses a diverse range of summarization tasks and includes comprehensive metadata, enabling the exploration of various research questions and supporting different applications. To showcase its utility, we conducted extensive evaluations using multiple state-of-the-art baselines. The results provide benchmarks for facilitating further research in this area.
Keywords: Corpus Resource, Summarization, Opinion Mining, Applications

1. Introduction

Advances in Pretrained Language Models (Raffel et al., 2020; Lewis et al., 2020; Zhang et al., 2020) and Large Language Models (Brown et al., 2020; Chowdhery et al., 2022; Workshop et al., 2022; Touvron et al., 2023) have propelled neural summarization to new heights. Existing research has primarily focused on standard summarization benchmarks within domains like news (Hermann et al., 2015; Narayan et al., 2018), dialogue (Gliwa et al., 2019), scientific articles (Cohan et al., 2018), and opinions (Angelidis and Lapata, 2018; Chu and Liu, 2019; Bražinskas et al., 2020). However, there is also a need for benchmarks that better represent real-life applications of summarization. These benchmarks should explore areas that have received limited attention and that present new and challenging use cases. By incorporating these underexplored domains into the evaluation process, we can effectively assess the performance of summarization models in scenarios where summarization can make a meaningful social impact. This paper addresses this need by introducing REFLECTSUMM, a novel dataset focusing on the summarization of 17,512 student reflections on 782 university lectures from 24 large STEM classes. Table 1 shows example reflections in response to a prompt regarding the interesting facets of a lecture. As suggested by Baird et al. (1991), reflections are useful for both students and teachers, enhancing their knowledge, self-awareness, and classroom practice. For example, providing reflection summaries can assist instructors in identifying key areas where students exhibit misconceptions, thereby enabling them to strategize appropriate follow-up actions for upcoming lectures (Fan et al., 2017). Compared to using human-crafted summaries, *automatic summarization can help scale the use of reflections in educational practice*.

It is important to recognize that *student reflections and their summaries differ from standard benchmark corpora* in the related area of opinion summarization,[1] which has traditionally focused on product and service reviews. Table 1 illustrates the *variability observed in the length and structure of reflections*. While some students opt for concise expressions using words or phrases, others delve deeper into the topic by composing complete sentences to highlight interesting lecture aspects. *Reflection summaries are also more abstractive* than standard opinion summaries (see Table 2, to be discussed below).

Furthermore, REFLECTSUMM *provides richer types of information compared to existing corpora* for summarizing both student reflections (Luo et al., 2016; Fan et al., 2017; Magooda and Litman, 2020) as well as opinions (Angelidis and Lapata, 2018; Bražinskas et al., 2020; Angelidis et al., 2021; Yang et al., 2023). While prior corpora emphasized either abstractive or extractive summarization, our dataset provides *three types of reference sum-*

* These authors contributed equally to this work.

[1] Opinions are similarly obtained from multiple humans and order doesn't matter.

Reflection Prompt
Describe what you found most interesting in today's class

Student Reflections
• Nothing in particular today -> 1.0
• Despite the confusion, I did find setting up these problems to be very interesting and rewarding. -> 3.0
• Equipotentials -> 2.0
• i thought the breakout room questions were interesting because i learned how to do questions -> 4.0
• I found the last problem in class the most interesting because it was proven we can derive almost anything. -> 4.0
• The most interesting thing was that finding electric potential doesn't require a path, but only the magnitude of the charge and it's distance from the point of interest. -> 4.0
• I really enjoy line integrals and I can tell that we're moving towards using them to calculate potential. -> 4.0
• Collection of point charges (pairing them) -> 2.0
• How we can calculate something so complicated as electrons passing through an area is very cool. -> 3.0
• I found equipotentials to be the most interesting thing, especially drawing a equipotentials for a dipole! -> 4.0
• I thought it was interesting that Vnet is equal to all Vs added together -> 4.0
• I found how conductors act to be interesting. -> 3.0
... *the rest is omitted to save space*

Abstractive Summary
The students today found calculations and relationships to other concepts that they have learned in this and other classes interesting. They also found potential energy and equipotentials very interesting, as well as some integration concepts.

Extractive Summary
• I found equipotentials to be the most interesting thing, especially ...
• The most interesting thing was that finding electric potential doesn't require a path, but only the magnitude of the charge and it ...
... three more extractive reflections omitted to save space

Phrase Summary
• equipotentials
• calculations
• relations to old concepts
• potential
• integration

Table 1: An example from the REFLECTSUMM dataset showing reflections annotated with specificity score (displayed after the special token "->") and three different types of reference summaries.

maries for each set of reflections: extractive, abstractive, and phrase-level extractive summaries. Additionally, we augment the dataset with valuable metadata, such as reflection *specificity scores*,[2] which can be used to improve summarization performance (see Section 7). Furthermore, we provide student *demographic information*, enabling the exploration of fairness and equity issues.

Our contributions can be summarized as follows: (1) We publicly release REFLECTSUMM, which contains 17,512 reflections on 782 lectures from 24 university courses, along with reference summaries and metadata, allowing for exploration and advancement in summarization. (2) We conduct a detailed analysis using both pretrained language models and large language models to benchmark the REFLECTSUMM dataset across abstractive, extractive, and phrase summarization tasks. (3) We investigate research directions leveraging the provided metadata by exploring the concept of *specificity-aware summarization*. The specificity metadata provides a way to integrate the study of specificity (Li and Nenkova, 2015; Gao et al., 2019) into summarization. Additionally, we showcase that our demographic information can assist further research in studying the fairness and bias problem in the context of summarization research (Sections 3.2 and 8). (4) We make our dataset, models, and model outputs publicly available at https://github.com/EngSalem/ReflectSUMM, enabling researchers to build upon our work.

2. Related Work

Prior student reflection datasets (Luo and Litman, 2015; Luo et al., 2016; Magooda and Litman, 2020) were constrained in their size, course diversity, and summarization task coverage (see Table 3, to be discussed below). Specifically, prior datasets not only summarized fewer lectures, but also covered fewer academic subjects and/or courses per subject, limiting investigations of how models generalize. In addition, only one of our three summarization tasks (extractive, abstractive, and phrase-based) is covered per prior work. Well-known review opinion summarization benchmarks are similarly constrained in their summarization task coverage, with OpoSum (Angelidis and Lapata, 2018) focused on extractive summarization and FewSumm (Bražinskas et al., 2020) instead focused on abstractive summarization (Table 2). REFLECTSUMM *provides reference summaries in three formats (abstractive, extractive, phrase-based), new types of metadata (reflection-level specificity annotations, student demographic information), and enables various evaluation scenarios (including but not limited to cross-course, within-course, course-agnostic, cross-subject, etc.).*

Most prior NLP work on student reflections has focused on quality (e.g. specificity) prediction (Kovanović et al., 2018; Ullmann, 2019; Carpenter et al., 2020). With respect to summarization, Luo and Litman (2015) suggested extracting noun

[2]Each reflection in Table 1 is assigned a score ranging from 1 to 4 (explained in Section 3.2).

phrases to compress the reflections for supporting mobile applications. Magooda and Litman (2020) utilized neural models with a focus on generating abstractive summaries in a low-resource context (Magooda et al., 2021; Magooda and Litman, 2021). *Our work follows the neural paradigm, further developing prior pretrained baselines and exploring the use of Large Language Models (LLMs).*

Our dataset can be considered as a special case of low-resource multi-document opinion summarization, where the opinions here refer to student reflections rather than service and product reviews (Angelidis and Lapata, 2018; Bražinskas et al., 2020). Most prior opinion work with limited data focused on synthesizing training data for intermediate finetuning (Bražinskas et al., 2020), parameter efficient techniques (Bražinskas et al., 2022), or second stage reranking (Oved and Levy, 2021). Previous low-resource work that targeted summarizing both student reflections and product reviews leveraged multitask learning with pretrained language models (Magooda et al., 2021), domain transfer from pretrained models (Magooda and Litman, 2020), and curriculum learning (Magooda and Litman, 2021). Recently, Large Language Models (Brown et al., 2020; Workshop et al., 2022; Sanh et al.; Touvron et al., 2023) have been explored for both news (Goyal et al., 2022; Zhang et al., 2024) and opinion (Bhaskar et al., 2023) summarization in zero-shot settings. *We provide several baseline results with both pretrained language models and LLMs to benchmark their utility in the zero-shot and one-shot summarization of reflective writing.*

3. REFLECTSUMM

3.1. Dataset Collection and Annotation

The student reflections in REFLECTSUMM were collected after each lecture in 24 courses from two American universities. The data were obtained across four semesters, from Fall 2020 to Spring 2022. Students used the CourseMirror Application (Fan et al., 2015)[3] to respond to two prompts: *(1) Describe what you found most interesting in today's class* and *(2) Describe what was confusing or needed more details in today's class*. These reflection prompts are based on learning sciences research, starting with Menekse et al. (2011), where students wrote reflections on paper and a TA manually summarized them. These prompts are polarity-specific (confusing versus interesting lecture aspects). Prior evaluations of early versions of the CourseMIRROR app used to collect our data (where the app only used phrase summarization at the time) found that reading the phrase summaries was viewed positively by both instructors and students (Fan et al., 2015). More recently, Menekse (2020) found that generating reflections and reading class phrase summaries improved student exam scores.

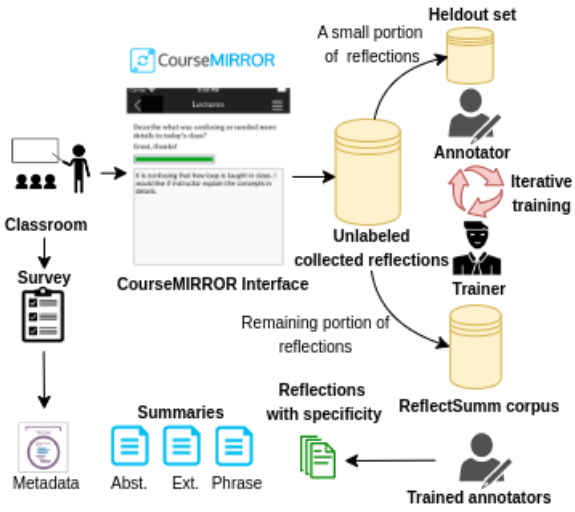


Figure 1: REFLECTSUMM corpus creation.

For the reflection quality annotation, following the guidelines in Luo and Litman (2016), annotators assigned a score from 1-4, where 4 means the reflection text has the highest specificity and 1 means the least specificity.

Turning to summarization, the phrase summary task for the annotator is to provide five phrases that can best summarize the students' reflections, together with how many students semantically mentioned each phrase. Those phrases can be either extracted from the reflections or manually constructed by the annotator. The annotators are further instructed to write an abstractive summary to summarize the major points of the full reflections. Lastly, the annotators select five reflections as the extractive summaries. Full annotation guidelines are in Appendix A.1.

Eleven college students with backgrounds in the appropriate subject domains were recruited to work on reflection scoring and summarization. Students were first trained on three batches of extra-held sets to understand and grasp the tasks before being assigned real jobs. The average pairwise inter-annotator agreement (IAA) across four students with double-annotations is 0.668 for the reflection score by Quadratic Weighted Kappa, suggesting substantial agreements. For summarization, we instead measure the averaged inter-annotator

[3]The application is downloadable from Apple Store https://apps.apple.com/us/app/coursemirror-v2/id1506495976 and Google Store https://play.google.com/store/apps/details?id=education.pittsburgh.cs.mips.cm_v2&hl=en_US&gl=US&pli=1

ROUGE scores (R-1/R-2/R-L) (Lin, 2004), which are 48.31/27.57/43.52 and 30.16/6.77/27.91 for the extractive and abstractive summarization tasks, respectively. These scores are slightly lower than those reported in Magooda (2022), which used the same guidelines but on different data. Figure 1 provides a summary of the annotation process involved in constructing REFLECTSUMM.[4]

3.2. Dataset Description and Details

REFLECTSUMM reflections were collected after 782 lectures from 24 courses spanning four different subjects: Engineering (ENGR), Physics (PHY), Computer Science (CS), and Computing Information (CMPINF).[5] The majority of both lectures (56.9%) and courses (54.2%) are from ENGR. The remaining data is fairly evenly distributed between CS (20.3% of lectures and 20.8% of courses) and PHY (18.2% of lectures and 16.7% of courses), with only a small percentage of CMPINF lectures and courses (4.6% and 8.3%, respectively).

A total of $17,512$ REFLECTSUMM reflections spanning the 782 lectures have been annotated for their specificity as noted above. The majority of the reflections (52.5%) were rated as having a specificity score of 3, indicating moderate specificity. 22.6% of the reflections received the highest rating of 4, while 14.2% received a score of 2 and only 10.7% received a score of 1. The average specificity score of sentences selected for extractive summaries is higher (3.08) compared to the discarded sentences (2.85), suggesting that considering specificity as additional information to guide summarization is worth exploring.

Before the data collection phase, students were asked to participate in a pre-survey that collects their demographic information. Male students make up the majority (55.71%), while the survey did not record 3.45% of students' gender information. We also observe a diverse distribution across multiple racial groups, and the majority (58.89%) are White followed by Asian (20.93%).

Table 2 compares REFLECTSUMM with several established multi-document opinion summarization datasets. **REFLECTSUMM** boasts 782 input-summary pairs, which represent the count of unique lectures featuring the "interesting/confusing" prompt in the summarization input. This dataset surpasses OPOSUM (Angelidis and Lapata, 2018) and FewSumm (Bražinskas et al., 2020) in terms of dataset size (column 2). While OPOSUM and FewSumm limit the number of documents per input, REFLECTSUMM has more variability in the number of words and documents per input (columns 3 and 4). Column 5 shows that our abstractive summarization task is more abstractive, as measured by the percentage of novel n-grams (See et al., 2017).[6] Lastly, REFLECTSUMM encompasses a distinctive blend of summarization tasks (column 6). Table 3 compares REFLECTSUMM with prior reflection-focused corpora. Columns 2-5 show that our dataset surpasses in the number of unique lectures featured with focused prompts, the breadth of courses covered, the diversity of reference summaries, and the availability of metadata.

4. Tasks and Benchmark Models

4.1. Extractive Summarization

Corresponding to the human extractive summary task, the goal of our models is to pinpoint the five most salient reflections (documents) from a collection of reflections within the same lecture. We evaluate several baseline models to benchmark extractive performance: the traditional unsupervised method, **LexRank** (Erkan and Radev, 2004); **BERTSUM-EXT** (Liu and Lapata, 2019), a pretrained language model crafted for extractive summarization; **MatchSum** (Zhong et al., 2020), a state-of-the-art summarization system which employs a re-ranker, and follows a two-stage paradigm to extract summaries; and **ChatGPT (GPT-3.5 turbo)**, a large language model capable of generating high-quality summaries. For ChatGPT, we devised two variants of prompts for experimentation in a zero-shot setting: **(**1**) GPT-reflect:** This variant prompts the model to select complete reflections, which mirrors how students typically write their reflections. **(**2**) GPT-reflect + specificity:** This variant integrates specificity scores with the original reflection and prompts the model to consider these scores while making selection choices. We also created a one-shot setting for the best-performing zero-shot model by randomly selecting a training split example and instructing the model to follow the example.

4.2. Extractive Phrase Summarization

The extractive phrase summarization task seeks to generate summaries constituted by five phrases, each supplemented with an accompanying number that indicates how many reflections support each phrase. This numerical task is inspired by the desire for a more comprehensible display of reflection distributions to instructors (Fan et al., 2015) while phrase extraction facilitates easier access on

[4] We justify the reason to select undergraduate students as annotators in Appendix A.2.

[5] ENGR courses come from a midwest US university, while the rest come from a northeast university.

[6] novelty = # new n-grams in summary/ # total n-grams in summary

Dataset	# Pairs	# Words/input (min/avg/max)	# Docs/input (min/avg/max)	Abstractiveness (1/2/3-grams)	Tasks
OPOSUM	600	468/485/499	10/10/10	-	Ext.
FEWSUMM Amazon	180	342/397/438	8/8/8	25.02/78.29/97.65	Abst.
FEWSUMM Yelp	300	362/399/433	8/8/8	26.04/80.71/98.76	Abst.
REFLECTSUMM	782	10/344/2229	4/22/79	36.97/83.11/98.12	Abst./Ext./Phrase

Table 2: Descriptive statistics comparing prior datasets (top) to REFLECTSUMM. **# Pairs** denotes the number of reflection/review document and summary pairs, while **# Words/Input (concatenated reflections/reviews)** represents the total word count in the input. **# of Docs (reflections/reviews)/input** indicates the number of documents per input . **Abstractiveness** measures abstractive summaries' novelty in terms of n-grams. (-) signifies that the information is not applicable for this dataset. **Tasks** include Abstractive (Abst.), Extractive (Ext.), and Phrase (Phr.) summarization.

Dataset	# Pairs	Course Coverage	Tasks	Metadata
Luo and Litman (2015)	36	1-ENGR	Phrase	No
Luo et al. (2016)	70	2-Statistics	Phrase	No
Magooda and Litman (2020)	188	4-ENGR/Statistics/CS	Abst.	No
REFLECTSUMM (ours)	782	24-ENGR/PHY/CS/CMPINF	Phrase/Abst./Ext.	Yes

Table 3: Comparison of **REFLECTSUMM** with previous reflection summarization datasets. **#Pairs** represents the count of unique lectures featuring the "interesting/confusing" prompt in the summarization input. **Course Coverage** describes the sources of reflections, presented in a count-course subjects format, **Tasks** delineates the types of reference summaries, and **Metadata** indicates the inclusion of additional information in the corpus (e.g., student demographics, reflection specificity).

mobile devices. To benchmark phrase summarization, we have employed the unsupervised model proposed by Luo and Litman (2015), aggregating noun phrases into five clusters and identifying the clusters' centroids as the phrase summary. We named it **PhraseSum**. In addition, we utilized OpenAI's **ChatGPT (GPT-3.5-turbo)** as our LLM to perform zero-shot phrase extraction. We experimented with **GPT-Human**, using the prompt provided to human annotators. However, as the original prompt doesn't specify the types of phrases to extract, we introduced two more baselines for additional experiments: (1) **GPT-noun phrase** aimed to extract just noun phrases for each lecture, and (2) **GPT-Human + noun** which adds "noun phrase" to the original human prompt. We also added a one-shot setting for the best-performing model, similar to the extractive summarization task. We include all prompts in Appendix B.

4.3. Abstractive Summarization

Human annotators were given the task of summarizing students' reflections concisely and coherently within 40 words. To benchmark this task, various models were employed, including fine-tuning pretrained language models, namely **BART-Large** (Lewis et al., 2020) and a modified version called **BART-Large+specificity**. The latter incorporates markers on a 4-point scale indicating reflection specificity scores, following the same approach used in literature for scientific articles (DeYoung et al., 2021), dialogue (Khalifa et al., 2021), and legal documents (Elaraby and Litman, 2022).[7]

As with the extractive models, we developed ChatGPT models too. In the zero-shot setting, we explored two prompting settings: **(**1**) GPT-Human:** using a version similar to the prompt given to human annotators and **(**2**) GPT-Human + specificity:** incorporating specificity scores of each reflection in the prompt. We also added a one-shot model as in the extractive summarization tasks.

5. Experimental Setup

All models are evaluated using cross-validation. Lectures are first grouped and shuffled by the subjects. Each subject is then divided into five folds by shuffling the lectures within that subject. We combine those four folds from each subject as the final training fold set and make the remaining test fold. We randomly select 10% of the data within each training fold set for validation and model selection.

We mainly evaluate our models using two standard metrics, ROUGE (Lin, 2004) and BERTScore (Zhang* et al., 2020). In addition, we report lecture-level reflection exact match F1 (EM F1) and partial

[7] See Appendix B.2 for an example of using markers to include specificity scores.

match F1 (P F1) scores for the extractive summarization task.[8] For the exact match F1, we compare the predicted and human-reference reflections on a per-lecture basis. Partial match F1 assesses the correctness of selecting partial components from a complete reflection, allowing for more flexibility in the evaluation. While standard extractive summarization tasks typically use human-written abstractive summaries as references, we utilize the annotated extractive reference summaries to evaluate the model outputs.

Abstractive models often suffer from hallucinations, generating information not present in the source text (Ji et al., 2023). To assess the factuality of our generated summaries, we utilize the pre-existing entailment metric called SUMMAC (Laban et al., 2022). This metric assesses the overall entailment score between the generated summary and the input document, considering different levels of granularity. The score can be computed by considering the entire document or computing the aggregated score from pair-wise sentence-level entailments. SUMMAC introduces two versions: $SummaC_{zs}$, which averages pairwise entailment scores. A score (ranging from 0 to 1) indicates a stronger alignment between the document and the summary, while a negative score (ranging from 0 to -1) suggests counterfactually-generated text; and $SummaC_{conv}$, where entailment scores are aggregated by a convolution layer to avoid mean sensitivity to extreme entailment values. The convolution layer aggregates values into 5 bins: $[0, 0.2)$, $[0.2, 0.4)$, $[0.4, 0.6)$, $[0.6, 0.8)$, and $[0.8, 1)$. A higher bin indicates a stronger factual consistency of the summary. We mainly relied on $SummaC_{conv}$ in our analysis, as recommended by the paper.[9]

6. Implementation Details

For the BERTSUM-EXT **extractive summarization** model, we leveraged a bertext_cnndm_transformer checkpoint that was trained on the CNN/DM news dataset using the original codebase[10] to select 5 reflections. We additionally fine-tuned BERT-EXT models on our dataset and the FEWSUMM AMAZON dataset to examine the benefits of our data for summarization tasks. We further experimented with BERT-EXT + specificity, where the specificity scores are incorporated into the input. For MatchSum, we used the checkpoint equipped with a RoBERTa-based re-ranker. We formed the candidate sets by employing the off-the-shelf BERT-EXT model to prune the original documents into 8 reflections and constructed the combinations of 5 sentences subject to the pruned document. For LexRank, we used the lexrank package,[11] treating the concatenation of all reflections from the train split of each fold as documents to initialize the model and setting the summary size at 5 with threshold ratio of 0.1. For ChatGPT, we utilized the OpenAI API.[12] We set the maximum tokens to 1024 and the temperature to 0.5.

We replicated the **extractive phrase summarization** model from Luo and Litman (2015), which utilizes KMedoid clustering of noun phrases extracted from reflections and encoded using the BERT-base model (details in Appendix D.1). For ChatGPT, we set the maximum tokens to 1024 and the temperature to 0.5.

We fine-tune the BART-Large **abstractive summarization** model for 10 epochs on each fold, employing an early stopping technique with the patience of 3 epochs. We utilize the HuggingFace implementation (Wolf et al., 2020). To identify the optimal model, we evaluate its R-2 score on the validation set. For the BART + reflection specificity tokens, we use human-annotated specificity scores during training and predicted specificity obtained by a finetuned-DistillBERT model. For LLMs, we use openAI's API and set the maximum tokens to 100 and the temperature to 0.7.

7. Results and Analysis

7.1. Extractive Summarization

Table 4 shows that the baseline BERTSUM-EXT is not as satisfactory as the traditional LexRank baseline. We observe that fine-tuning the model on our specific dataset brings appreciable performance gains, as evidenced by the comparison between row 2 and row 4. Furthermore, fine-tuning on a similar opinion summarization dataset also enhances performance, though not to the same extent as using our in-domain data (row 2 vs. row 3). Including specificity scores in the dataset brings improvements in R-2 and helps with the matching F1 when compared to the fine-tuned baseline without specificity information (row 4 vs. row 5). The state-of-the-art model MatchSum obtained the second-best performances regarding ROUGE

[8] See Appendix C.1 for an illustrative example of how these scores are computed.

[9] We opted against using factuality metrics based on question-answering (QA) approaches like QAFactEval (Fabbri et al., 2022). This decision was due to limitations in entity extraction, which struggled to recognize educational concepts and noisy noun phrase generation caused by the varied structure of input reflections, as shown in the example included in Appendix C.2.

[10] https://github.com/nlpyang/PreSumm

[11] https://github.com/crabcamp/lexrank

[12] https://platform.openai.com/docs/api-reference

Model	R-1	R-2	R-L	BS	EM F1	P F1
LexRank	56.96	40.45	55.10	89.95	31.33	37.16
BERTSUM-EXT (cnndm)	55.21	38.29	53.39	89.91	34.08	37.74
BERTSUM-EXT (ft. FEWSUMM AMAZON)	55.51	38.41	53.59	89.88	33.85	37.81
BERTSUM-EXT (ft. REFLECTSUMM)	56.15	39.09	54.25	89.94	33.14	37.49
BERTSUM-EXT (ft. REFLECTSUMM) + Specificity	55.94	39.50	54.16	89.31	33.45	37.84
MatchSum	58.79*	42.59*	56.70*	**90.57**	**36.26***	**38.94**
GPT-reflect	**60.16***	**43.93***	**58.26***	89.98	21.41	37.68
GPT-reflect + specificity	58.76*	42.49*	56.85*	90.29	21.28	36.45
GPT-reflect - one-shot	58.65	41.04	56.46	89.58	20.18	33.07

Table 4: Extractive summarization model performance reported on ROUGE (R-1, R-2, R-L), BERTScore (BS), Exact Match F1 (EM F1) and Partial F1 (P F1). The best column results are **bolded**, while * means statistically different from the baseline LexRank (p-value < 0.05) using a paired t-test.

Model	R-1	R-2	R-L	BS
PhraseSum	24.87	7.98	24.31	83.9
GPT-Human	34.25*	11.25*	33.27*	84.7*
GPT-noun phrase	39.28*	14.55*	38.26*	87.1*
GPT-Human + noun	38.86*	13.56*	38.02*	84.6*
GPT-noun - one-shot	**40.43***	**15.48***	**39.51***	**87.7***

Table 5: Extractive phrase summarization model performance. The best result of each column is **bold**. * means statistically different from the baseline PhraseSum (p-value < 0.05) using a paired t-test.

scores and the highest BERTScore and matching performances.[13] Meanwhile, ChatGPT models obtain the best or on-par performance concerning ROUGE and BERTScores. However, it should be noted that the ChatGPT-based models struggle to fully extract the reflections, as evidenced by the lower Exact Match F1 scores. We posit the improvements from Exact to Partial F1 are attributed to the incapability of ChatGPT models to comprehend the prompt fully, thus cutting the original reflections into sentences and making partial selections. Overall, the Partial F1 score suggests that there is still ample room to improve the extractive summarization models to match human performance. Based on the best zero-shot setting (*GPT-reflect*), we also explored the one-shot setting and found no gains.

To validate the hypothesis that ChatGPT-based models may not be able to perform the extractive task faithfully, we analyzed the proportion of model output sentences that are fully extracted from the original reflections. In detail, we measure the ratio of the system-extracted reflections that come from the original reflections instead of being generated creatively by the ChatGPT model. Compared to the BERTSUM-EXT model with near-perfect extractiveness (99.4%),[14] ChatGPT models obtained 92.53, and 94.68% extractiveness scores for *GPT-reflect* and *GPT-reflect + specificity*, respectively, showing that GPT models do sometimes generate non-extractive sentence/reflections. In contrast, MatchSum follows a two-stage paradigm to extract summaries. The output reflections are guaranteed to be selected from the original reflections, securing higher EM F1 and P F1.

7.2. Extractive Phrase Summarization

Table 5 shows the results.[15] The PhraseSum model obtains the worst performance. We observe a difference when adjusting the prompt for the ChatGPT models. Comparing *GPT-Human* and *GPT-Human + noun* shows that just replacing "phrases" with "noun phrases" brings about 4.5 points improvements on R-1 and R-L, 2.3 points on R-2. Meanwhile, *GPT-noun phrase* obtains improvements of 5.0, 3.3, and 4.9 ROUGE (1,2, L) scores compared to *GPT-Human*. We posit that the relatively lower performance of *GPT-Human* compared to *GPT-noun phrase* is that, in *GPT-Human*, the task is inherently a multi-task project by adding the prompt of *together with how many students semantically mentioned each phrase in parenthesis*. Finally, based on the best zero-shot setting, we explored a one-shot setting for *GPT-noun phrase*, where a random sample from the data was used. Unlike in Table 4, one-shot outperforms all zero-shot settings, perhaps because phrase samples are more consistent than the full reflections.

[13] Example outputs for all models are in Appendix E.

[14] The score does not reach 100% since some annotators selected sentences instead of full reflections.

[15] We apply regular expressions to clean up the GPT model outputs. Details are in Appendix D.3.

Model	R-1	R-2	R-L	BS
BART-Large	47.09	24.17	43.76	90.49
+ specificity	**47.70**	**24.85**	**44.41***	**90.57**
GPT-Human	35.83	9.40	31.85	88.23
+ specificity	36.73	9.13	31.64	88.27
GPT-one-shot	36.86	9.46	31.96	88.26

Table 6: Abstractive summarization performance. Best column result **bolded**; * is statistically different from the baseline BART-Large.

Model	%Novel 1/2/3 grams	Length min/avg/max
Human-refer.	37.00/83.18/98.16	23/46/99
BART-Large	36.91/79.67/95.11	28/45/85
+ specificity	35.27/77.29/93.18	26/42/85
GPT-Human	27.71/74.65/94.40	19/35/68
+ specificity	29.10/74.19/94.02	15/38/66
GPT-one-shot	31.43/77.59/95.39	15/35/66

Table 7: Percent of novel n-grams and length statistics in abstractive summaries.

7.3. Abstractive Summarization

We report our abstractive summarization results in Table 6. Our results demonstrate that incorporating *specificity markers* into the input achieved the best fine-tuned BART baseline performance by a small margin (rows 1 vs. 2; $p < 0.05$ for R-L). Unlike the results for the prior two extractive summarization tasks, none of the GPT-based LLMs could match the performance of more traditional methods, with respect to ROUGE and BERTScore. To examine whether characteristics of the LLM summary output might be a factor in their poor ROUGE performance, Table 7 shows the percentage of novel n-grams present in each summary, as well as the average summary length. The novelty figures indicate that LLMs generally have a lower proportion of novel n-grams compared to fine-tuned models, which maximize generating summaries that follow human-written summaries (high in novelty). However, incorporating one-shot learning improves n-gram novelty in LLMs, showing that providing even one example enhances the generation of novel outputs. Additionally, the abstractive summaries generated by LLMs are generally shorter than human-written summaries and those produced by fine-tuned models, potentially contributing to lower ROUGE.

Finally, Table 8 shows the results of the factuality evaluation using SUMMAC. For computing off-the-shelf entailment scores, we utilize the AlBERT model, which was fine-tuned on an entailment dataset as described in Schuster et al. (2021).

Model	SUMMAC ↑ Sentence	SUMMAC ↑ Document
Human-reference	0.25	0.22
BART-Large	0.25	0.21
+ specificity	0.25	0.22
GPT-Human	0.26	**0.31**
+ specificity	**0.27**	0.26
GPT-one-shot	0.26	0.26

Table 8: Factuality scores based on SUMMAC(↑: higher means better entailment).

Our results indeed highlight the challenge and limitations of current factuality metrics when applied to this new type of data. The results indicate that *GPT-Human* demonstrates the highest level of overall agreement with the input reflections, surpassing even the human-written summaries at both the sentence-level (*GPT-Human +specificity*) and document-level resolutions. This finding is surprising since we guaranteed the quality of our human-written summaries by providing annotator training, as detailed in Section 3.1. Therefore, a higher average SummaC score for the summaries generated by chatGPT does not necessarily indicate a more factual summary when compared to the human-written ones.[16]

We postulate that this disparity arises as our reflections may contain individual words or phrases rather than complete sentences, thereby deviating from the training data of the entailment model, which consists of complete sentences (Williams et al., 2018). Therefore, a thorough qualitative analysis is crucial to assess the factual accuracy of generated summaries. Novel factuality metrics tailored to reflective writing and similar summarization domains are promising avenues for future research.

8. Broader Impact of the Dataset

This paper focused on introducing and utilizing the REFLECTSUMM dataset to develop and evaluate benchmark models for three summarization tasks. For the NLP community, the dataset can enable the creation of new benchmarks for other tasks by harnessing the rich metadata to be released with the dataset. For instance, researchers can use the student demographics to work on analyses of potential fairness or equity issues (Dash et al., 2019) in summarization and build fairness-oriented summarization models. Initial explorations indicate that REFLECTSUMM shows promise for this purpose, as it reveals a difference in the distribution

[16] See Appendix C.3 for an illustrative example.

of reflections along the gender dimension between the extractive summaries and the entire dataset.[17]

Moreover, new ways to use the manually annotated specificity annotations beyond those presented here can bridge summarization and NLP research on specificity prediction more broadly (Li and Nenkova, 2015; Gao et al., 2019). Also, our tasks and models could enable new downstream functionalities in educational technologies already collecting reflections (Fan et al., 2015; Carpenter et al., 2021) such as generating recommended readings and explaining confusing concepts based on summary output. Learning scientists may also find our dataset valuable for monitoring the growth of students across a semester by analyzing reflections for the same course over time. On average, students contributed course reflections to 42 percent of the lectures throughout the semester. The longitudinal aspect of our data might also be used to define new summarization tasks.

9. Conclusion and Future Work

We present REFLECTSUMM, a new dataset designed for course reflection summarization that includes multiple summarization tasks. The dataset provides specificity annotations of reflections and metadata on user demographics. To demonstrate its utility, we benchmarked the dataset. Our results demonstrated how the benefits of pretrained and finetuned language models, large language models, reflection specificity, and one-shot learning techniques could vary significantly across different summarization tasks, shedding light on the nuanced advantages of these approaches.

Future Work We foresee numerous future directions that can be built upon the dataset's rich information coverage. For instance, we plan to enhance the efficacy of LLMs by investigating improved prompting techniques and developing more appropriate evaluation metrics that align with the nature of students' reflective writing. Our benchmark results also uncover that prompts modeled on human summarization guidelines are insufficient and that it remains challenging to incorporate best supervised examples and/or specificity into the summarization tasks. Additionally, extending our dataset across different domains is necessary and beneficial for future research. We have collected data from two psychology courses (new domain) from a Canadian institute (outside the US) and a few Math and Mechanical Engineering courses from US institutes. We plan to add those as future additions to the REFLECTSUMM dataset. The number of extractive phrases or reflections should be dynamically adjusted for the extractive summarization tasks, as real-world scenarios vary across different lectures and subjects. Our analysis reveals 75 of the 782 lectures have 10 or fewer students. In these cases, the number of shared topics can be fewer than 5. One future direction would be a dynamic system that utilizes topic models to handle different-sized reflection collections.

[17] There are more male-written reflections in extractive summaries (57%) than in the overall distribution (52%), verified with chi-square test.

Limitations

For experiments involving the recent advanced LLMs, we created fairly straightforward benchmarks. Future work will need investigations of methods such as prompt optimization to better unleash the capability of ChatGPT and other LLMs (Qin et al., 2023; Bang et al., 2023). More supervised extractive/abstractive summarization models are yet to be added, and we encourage researchers to help contribute to the benchmarks. We also acknowledge that the released dataset may be of a narrow scope of subjects, including only engineering and science-related courses. With the released protocol, we envision extending the dataset to cover courses from the liberal arts and other backgrounds and facilitate the overall quality of teaching and education. We also have not yet incorporated human assessments to gauge the quality of various system outputs and explore the similarities or differences between human-authored and AI-generated outputs. In this resource paper, our primary focus is to introduce the details of the proposed dataset alongside showcasing its tasks through benchmarking experiments carried out on prevailing domain-specific models. Our experimental exploration, though not as focused on the intricacies of modeling as typically seen in modeling-based summarization papers, offers a comprehensive insight into the practical performance and applicability of different models. It establishes a robust foundation for further investigations into course reflection summarization.

Data Statement

We are dedicated to making the dataset, including metadata such as demographic information and specificity scores, accessible to the public. Furthermore, we pledge to provide supplementary materials instrumental in crafting this dataset and conducting experiments. These materials, such as the **annotation guidelines**, the **list of prompts** used for LLM experiments with GPT, and **examples of our evaluation metrics**, can be accessible at https://github.com/EngSalem/ReflectSUMM and in the Appendices.

Ethics Statement

Safely using user demographic information becomes increasingly important in the current era. We carefully filtered the metadata that may leak the student's personal information, including emails, first and last names, and other attributes in the released version. We also only included data from students who explicitly provided consent.

Acknowledgement

The research reported here was supported in part by the Institute of Education Sciences, U.S. Department of Education, through Grant R305A180477, and the National Science Foundation through Grants 2329273 and 2329274. The opinions expressed are those of the authors and do not represent the views of the U.S. Department of Education or the National Science Foundation. We want to thank the members of the Pitt PETAL group, Pitt NLP group, and anonymous reviewers for their valuable comments in improving this work.

10. References

Stefanos Angelidis, Reinald Kim Amplayo, Yoshihiko Suhara, Xiaolan Wang, and Mirella Lapata. 2021. Extractive opinion summarization in quantized transformer spaces. *Transactions of the Association for Computational Linguistics*, 9:277–293.

Stefanos Angelidis and Mirella Lapata. 2018. Summarizing opinions: Aspect extraction meets sentiment prediction and they are both weakly supervised. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3675–3686.

John Baird, Peter Fensham, Richard Gunstone, and Richard White. 1991. The importance of reflection in improving science teaching and learning. *Journal of Research in Science Teaching*, 28:163 – 182.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity.

Adithya Bhaskar, Alex Fabbri, and Greg Durrett. 2023. Prompted opinion summarization with GPT-3.5. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9282–9300, Toronto, Canada. Association for Computational Linguistics.

Arthur Bražinskas, Mirella Lapata, and Ivan Titov. 2020. Few-shot learning for opinion summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4119–4135.

Arthur Bražinskas, Ramesh Nallapati, Mohit Bansal, and Markus Dreyer. 2022. Efficient few-shot fine-tuning for opinion summarization. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1509–1523.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Dan Carpenter, Elizabeth Cloude, Jonathan Rowe, Roger Azevedo, and James Lester. 2021. Investigating student reflection during game-based learning in middle grades science. In *LAK21: 11th International Learning Analytics and Knowledge Conference*, LAK21, page 280–291, New York, NY, USA. Association for Computing Machinery.

Dan Carpenter, Michael Geden, Jonathan Rowe, Roger Azevedo, and James Lester. 2020. Automated analysis of middle school students' written reflections during game-based learning. In *AIED 2020: Artificial Intelligence in Education*, pages 67–78.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Eric Chu and Peter Liu. 2019. Meansum: A neural model for unsupervised multi-document abstractive summarization. In *International Conference on Machine Learning*, pages 1223–1232. PMLR.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 615–621.

Abhisek Dash, Anurag Shandilya, Arindam Biswas, Kripabandhu Ghosh, Saptarshi Ghosh, and Abhijnan Chakraborty. 2019. Summarizing user-generated textual content: Motivation and methods for fairness in algorithmic summaries. *Proc. ACM Hum.-Comput. Interact.*, 3(CSCW).

Jay DeYoung, Iz Beltagy, Madeleine van Zuylen, Bailey Kuehl, and Lucy Wang. 2021. Ms^2: Multi-document summarization of medical studies. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7494–7513.

Mohamed Elaraby and Diane Litman. 2022. Arglegalsumm: Improving abstractive summarization of legal documents with argument mining. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6187–6194.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479.

Alexander Richard Fabbri, Chien-Sheng Wu, Wenhao Liu, and Caiming Xiong. 2022. QaFactEval: Improved qa-based factual consistency evaluation for summarization. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2587–2601.

Xiangmin Fan, Wencan Luo, Muhsin Menekse, Diane Litman, and Jingtao Wang. 2015. CourseMIRROR: Enhancing large classroom instructor-student interactions via mobile interfaces and natural language processing. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '15, page 1473–1478, New York, NY, USA. Association for Computing Machinery.

Xiangmin Fan, Wencan Luo, Muhsin Menekse, Diane Litman, and Jingtao Wang. 2017. Scaling reflection prompts in large classrooms via mobile interfaces and natural language processing. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces*, IUI '17, page 363–374, New York, NY, USA. Association for Computing Machinery.

Yifan Gao, Yang Zhong, Daniel Preoţiuc-Pietro, and Junyi Jessy Li. 2019. Predicting and analyzing language specificity in social media posts. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6415–6422.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. *EMNLP-IJCNLP 2019*, page 70.

Tanya Goyal, Junyi Jessy Li, and Greg Durrett. 2022. News summarization and evaluation in the era of gpt-3. *arXiv preprint arXiv:2209.12356*.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 1693–1701, Cambridge, MA, USA. MIT Press.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12).

Muhammad Khalifa, Miguel Ballesteros, and Kathleen Mckeown. 2021. A bag of tricks for dialogue summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8014–8022.

Vitomir Kovanović, Srećko Joksimović, Negin Mirriahi, Ellen Blaine, Dragan Gašević, George Siemens, and Shane Dawson. 2018. Understand students' self-reflections through learning analytics. In *Proceedings of the 8th International Conference on Learning Analytics and Knowledge*, LAK '18, page 389–398, New York, NY, USA. Association for Computing Machinery.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. 2022. SummaC: Re-visiting NLI-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Junyi Jessy Li and Ani Nenkova. 2015. Fast and accurate prediction of sentence specificity. In

Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI'15, page 2281–2287. AAAI Press.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3730–3740, Hong Kong, China. Association for Computational Linguistics.

Wencan Luo and Diane Litman. 2015. Summarizing student responses to reflection prompts. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1955–1960, Lisbon, Portugal. Association for Computational Linguistics.

Wencan Luo and Diane J. Litman. 2016. Determining the quality of a student reflective response. In *Proceedings of the Twenty-Ninth International Florida Artificial Intelligence Research Society Conference*, pages 226–231.

Wencan Luo, Fei Liu, Zitao Liu, and Diane Litman. 2016. Automatic summarization of student course feedback. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 80–85, San Diego, California. Association for Computational Linguistics.

Ahmed Magooda. 2022. Techniques to enhance abstractive summarization model training for low resource domains. PhD Thesis.

Ahmed Magooda and Diane Litman. 2021. Mitigating data scarceness through data synthesis, augmentation and curriculum for abstractive summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2043–2052.

Ahmed Magooda, Diane Litman, Ahmed Ashraf, and Muhsin Menekse. 2022. Improving the quality of students' written reflections using natural language processing: Model design and classroom evaluation. In *Artificial Intelligence in Education: 23rd International Conference, AIED 2022, Durham, UK, July 27–31, 2022, Proceedings, Part I*, page 519–525, Berlin, Heidelberg. Springer-Verlag.

Ahmed Magooda, Diane Litman, and Mohamed Elaraby. 2021. Exploring multitask learning for low-resource abstractive summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1652–1661, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ahmed Magooda and Diane J. Litman. 2020. Abstractive summarization for low resource data using domain transfer and data synthesis. In *The Thirty-Third International FLAIRS Conference (FLAIRS-33)*, pages 240–245.

Muhsin Menekse. 2020. The reflection-informed learning and instruction to improve students' academic success in undergraduate classrooms. *The Journal of Experimental Education*, 88(2):183–199.

Muhsin Menekse, Glenda Stump, Stephen Krause, and M.T.H. Chi. 2011. The effectiveness of students' daily reflections on learning in engineering context. *ASEE Annual Conference and Exposition, Conference Proceedings*.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807.

Nadav Oved and Ran Levy. 2021. PASS: Perturb-and-select summarizer for product reviews. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 351–365, Online. Association for Computational Linguistics.

Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. 2023. Is ChatGPT a general-purpose natural language processing task solver? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1339–1384, Singapore. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled

version of bert: smaller, faster, cheaper and lighter. *The 5th EMC2 - Energy Efficient Training and Inference of Transformer Based Models, co-located at NeurIPS 2019*.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, et al. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*.

Tal Schuster, Adam Fisch, and Regina Barzilay. 2021. Get your vitamin C! robust fact verification with contrastive evidence. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 624–643, Online. Association for Computational Linguistics.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Thomas Ullmann. 2019. Automated analysis of reflection in writing: Validating machine learning approaches. *International Journal of Artificial Intelligence in Education*, 29.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

BigScience Workshop, :, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien, David Ifeoluwa Adelani, Dragomir Radev, Eduardo González Ponferrada, Efrat Levkovizh, Ethan Kim, Eyal Bar Natan, Francesco De Toni, Gérard Dupont, Germán Kruszewski, Giada Pistilli, Hady Elsahar, Hamza Benyamina, Hieu Tran, Ian Yu, Idris Abdulmumin, Isaac Johnson, Itziar Gonzalez-Dios, Javier de la Rosa, Jenny Chim, Jesse Dodge, Jian Zhu, Jonathan Chang, Jörg Frohberg, Joseph Tobing, Joydeep Bhattacharjee, Khalid Almubarak, Kimbo Chen, Kyle Lo, Leandro Von Werra, Leon Weber, Long Phan, Loubna Ben allal, Ludovic Tanguy, Manan Dey, Manuel Romero Muñoz, Maraim Masoud, María Grandury, Mario Šaško, Max Huang, Maximin Coavoux, Mayank Singh, Mike Tian-Jian Jiang, Minh Chien Vu, Mohammad A. Jauhar, Mustafa Ghaleb, Nishant Subramani, Nora Kassner, Nurulaqilla Khamis, Olivier Nguyen, Omar Espejel, Ona de Gibert, Paulo Villegas, Peter Henderson, Pierre Colombo, Priscilla Amuok, Quentin Lhoest, Rheza Harliman, Rishi Bommasani, Roberto Luis López, Rui Ribeiro, Salomey Osei, Sampo Pyysalo, Sebastian Nagel, Shamik Bose, Shamsuddeen Hassan Muhammad, Shanya Sharma, Shayne Longpre, Somaieh Nikpoor, Stanislav Silberberg, Suhas Pai, Sydney Zink, Tiago Timponi Torrent, Timo Schick, Tristan Thrush, Valentin Danchev, Vassilina Nikoulina, Veronika Laippala, Violette Lepercq, Vrinda Prabhu, Zaid Alyafeai, Zeerak Talat, Arun Raja, Benjamin Heinzerling, Chenglei Si, Davut Emre Taşar, Elizabeth Salesky, Sabrina J. Mielke, Wilson Y. Lee, Abheesht Sharma, Andrea Santilli, Antoine Chaffin, Arnaud Stiegler, Debajyoti Datta, Eliza Szczechla, Gunjan Chhablani, Han Wang, Harshit Pandey, Hendrik Strobelt, Jason Alan Fries, Jos Rozen,

Leo Gao, Lintang Sutawika, M Saiful Bari, Maged S. Al-shaibani, Matteo Manica, Nihal Nayak, Ryan Teehan, Samuel Albanie, Sheng Shen, Srulik Ben-David, Stephen H. Bach, Taewoon Kim, Tali Bers, Thibault Fevry, Trishala Neeraj, Urmish Thakker, Vikas Raunak, Xiangru Tang, Zheng-Xin Yong, Zhiqing Sun, Shaked Brody, Yallow Uri, Hadar Tojarieh, Adam Roberts, Hyung Won Chung, Jaesung Tae, Jason Phang, Ofir Press, Conglong Li, Deepak Narayanan, Hatim Bourfoune, Jared Casper, Jeff Rasley, Max Ryabinin, Mayank Mishra, Minjia Zhang, Mohammad Shoeybi, Myriam Peyrounette, Nicolas Patry, Nouamane Tazi, Omar Sanseviero, Patrick von Platen, Pierre Cornette, Pierre François Lavallée, Rémi Lacroix, Samyam Rajbhandari, Sanchit Gandhi, Shaden Smith, Stéphane Requena, Suraj Patil, Tim Dettmers, Ahmed Baruwa, Amanpreet Singh, Anastasia Cheveleva, Anne-Laure Ligozat, Arjun Subramonian, Aurélie Névéol, Charles Lovering, Dan Garrette, Deepak Tunuguntla, Ehud Reiter, Ekaterina Taktasheva, Ekaterina Voloshina, Eli Bogdanov, Genta Indra Winata, Hailey Schoelkopf, Jan-Christoph Kalo, Jekaterina Novikova, Jessica Zosa Forde, Jordan Clive, Jungo Kasai, Ken Kawamura, Liam Hazan, Marine Carpuat, Miruna Clinciu, Najoung Kim, Newton Cheng, Oleg Serikov, Omer Antverg, Oskar van der Wal, Rui Zhang, Ruochen Zhang, Sebastian Gehrmann, Shachar Mirkin, Shani Pais, Tatiana Shavrina, Thomas Scialom, Tian Yun, Tomasz Limisiewicz, Verena Rieser, Vitaly Protasov, Vladislav Mikhailov, Yada Pruksachatkun, Yonatan Belinkov, Zachary Bamberger, Zdeněk Kasner, Alice Rueda, Amanda Pestana, Amir Feizpour, Ammar Khan, Amy Faranak, Ana Santos, Anthony Hevia, Antigona Unldreaj, Arash Aghagol, Arezoo Abdollahi, Aycha Tammour, Azadeh HajiHosseini, Bahareh Behroozi, Benjamin Ajibade, Bharat Saxena, Carlos Muñoz Ferrandis, Daniel McDuff, Danish Contractor, David Lansky, Davis David, Douwe Kiela, Duong A. Nguyen, Edward Tan, Emi Baylor, Ezinwanne Ozoani, Fatima Mirza, Frankline Ononiwu, Habib Rezanejad, Hessie Jones, Indrani Bhattacharya, Irene Solaiman, Irina Sedenko, Isar Nejadgholi, Jesse Passmore, Josh Seltzer, Julio Bonis Sanz, Livia Dutra, Mairon Samagaio, Maraim Elbadri, Margot Mieskes, Marissa Gerchick, Martha Akinlolu, Michael McKenna, Mike Qiu, Muhammed Ghauri, Mykola Burynok, Nafis Abrar, Nazneen Rajani, Nour Elkott, Nour Fahmy, Olanrewaju Samuel, Ran An, Rasmus Kromann, Ryan Hao, Samira Alizadeh, Sarmad Shubber, Silas Wang, Sourav Roy, Sylvain Viguier, Thanh Le, Tobi Oyebade, Trieu Le, Yoyo Yang, Zach Nguyen, Abhinav Ramesh Kashyap, Alfredo Palasciano, Alison Callahan, Anima Shukla, Antonio Miranda-Escalada, Ayush Singh, Benjamin Beilharz, Bo Wang, Caio Brito, Chenxi Zhou, Chirag Jain, Chuxin Xu, Clémentine Fourrier, Daniel León Periñán, Daniel Molano, Dian Yu, Enrique Manjavacas, Fabio Barth, Florian Fuhrimann, Gabriel Altay, Giyaseddin Bayrak, Gully Burns, Helena U. Vrabec, Imane Bello, Ishani Dash, Jihyun Kang, John Giorgi, Jonas Golde, Jose David Posada, Karthik Rangasai Sivaraman, Lokesh Bulchandani, Lu Liu, Luisa Shinzato, Madeleine Hahn de Bykhovetz, Maiko Takeuchi, Marc Pàmies, Maria A Castillo, Marianna Nezhurina, Mario Sänger, Matthias Samwald, Michael Cullan, Michael Weinberg, Michiel De Wolf, Mina Mihaljcic, Minna Liu, Moritz Freidank, Myungsun Kang, Natasha Seelam, Nathan Dahlberg, Nicholas Michio Broad, Nikolaus Muellner, Pascale Fung, Patrick Haller, Ramya Chandrasekhar, Renata Eisenberg, Robert Martin, Rodrigo Canalli, Rosaline Su, Ruisi Su, Samuel Cahyawijaya, Samuele Garda, Shlok S Deshmukh, Shubhanshu Mishra, Sid Kiblawi, Simon Ott, Sinee Sang-aroonsiri, Srishti Kumar, Stefan Schweter, Sushil Bharati, Tanmay Laud, Théo Gigant, Tomoya Kainuma, Wojciech Kusa, Yanis Labrak, Yash Shailesh Bajaj, Yash Venkatraman, Yifan Xu, Yingxin Xu, Yu Xu, Zhe Tan, Zhongli Xie, Zifan Ye, Mathilde Bras, Younes Belkada, and Thomas Wolf. 2022. BLOOM: A 176B-Parameter Open-Access Multilingual Language Model. *arXiv e-prints*, page arXiv:2211.05100.

Xianjun Yang, Kaiqiang Song, Sangwoo Cho, Xiaoyang Wang, Xiaoman Pan, Linda Petzold, and Dong Yu. 2023. OASum: Large-scale open domain aspect-based summarization. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4381–4401, Toronto, Canada. Association for Computational Linguistics.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B. Hashimoto. 2024. Benchmarking Large Language Models for News Summarization. *Transactions of the Association for Computational Linguistics*, 12:39–57.

Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. Extractive summarization as text matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6197–6208, Online. Association for Computational Linguistics.

A. Human Annotation Details

A.1. Annotation Guidelines

Figure 2 shows the guideline we used for reflection specificity score annotation. We provided the original human annotation guidelines for summarization tasks in Figures 3 to 6.

A.2. Rationales on Human Annotator Selection

We worked with undergraduate students on this task for continuity and practicality. First, in prior motivating work from the learning sciences (Menekse et al., 2011) and as noted in Section 3.1, course TAs created summaries, so we wanted to keep a similar model, and we wanted to evaluate to what degree the NLP-generated summaries are similar to human-generated summaries. Second, practically, undergraduate students are more readily available and easier to recruit than course instructors. Importantly, our selection criteria for these individuals were not arbitrary; we chose students based on their academic majors. This ensured that they were familiar with the course content and had successfully met the course requirements themselves.

It would be interesting to ask some experienced professionals to do the same task and compare the results between raters. This comparison could yield a richer understanding of summarization quality and style variances across different expertise levels.

B. ChatGPT Prompts and BART's Markers

B.1. Extractive Prompts

We include the prompts for the extractive summarization task in Table 9, and the prompts for the extractive phrase summarization in Table 10.

B.2. Abstractive BART + markers Example

During training, we use oracle markers, while we rely on model predictions during inference. We include an example in Table 11.

B.3. Abstractive Summarization Prompts

To ensure a fair comparison with models that utilized BART-large, which underwent fine-tuning on reference summaries generated by human annotators, where the annotators were instructed to create approximately 40 word summaries, we followed a similar approach when providing instructions to GPT-Human. This was done to maintain consistency in the experimental setup and enable a meaningful comparison between the two models. For the one-shot setting, we rely on in-context learning by providing an example from the reference summaries annotated by human annotators. Table 12 shows the prompts we used in each setting.

Model	Prompt
GPT-reflect	"reflections: {{reflections}} Can you summarize the reflections, which are split by the special token ||_||, by extracting and selecting 5 original reflections from the split list?"
GPT-reflect + specificity	"reflections: {{reflections}} Can you summarize the reflections, which are split by the special token ||_||. Each reflection ends with a special marker -> and the specificity score in a range of 1-4, where 1 is the least specific and 4 is the most specific. Can you extract and select 5 original reflections from the split list by removing the ending "->" with the specificity score?"

Table 9: ChatGPT Prompts used for the extractive summarization task.

Model	Prompt
GPT-Human	"reflections: {{reflections}} Create a summary using five phrases together with how many students semantically mentioned each phrase in parenthesis. You can use your own phrases."
GPT-noun phrase	"reflections: {{reflections}} Can you summarize the reflections by extracting and selecting five noun phrases?"
GPT-Human + noun	"reflections: {{reflections}} Create a summary using five **noun phrases** together with how many students semantically mentioned each phrase in parenthesis. You can use your own phrases."

Table 10: ChatGPT Prompts used for the extractive phrase summarization task.

Example
<high> Energy conservation in particle physics with a gamma ray photon being split into an ... since I've never learned much about gamma rays in the past. </high> "\n" <bad> Nothing in particular today </bad>

Table 11: An example of using markers for quality in our dataset, and all reflections are concatenated with new line symbols.

Model	Prompt
GPT-Human	"reflections: {{reflections}}} Given the students' responses, create a short summary with no more than 40 words"
GPT-Human + specificity	"reflections: {{reflections}}} Each reflection ends with a special marker -> and the specificity score in a range of 1-4, where 1 is the least specific and 4 is the most specific. Given the students' responses, create a short summary with no more than 40 words, don't include the specificity scores in the summary."
GPT-oneshot	"reflections: {{oneshot reflections}} summary: {{oneshot summary}} reflections: {{test reflections}} summary:"

Table 12: ChatGPT Prompts used for the abstractive summarization task.

C. Evaluation Metrics

C.1. Extractive F1 Metrics

Besides the classical evaluation metrics including ROUGE and BERTScore, we additionally report the reflection-level exact match F1 and partial F1s. The intuition is to measure the selection accuracy of a given system provided with five human-annotated choices. As shown in Table 13, the GPT models can sometimes extract a portion of the original reflections, even with a carefully designed prompt. Using a too-strict exact matching would make the F1 score unfaithfully low, so we instead consider the partial matching on the sentence level and compute the partial match F1 accordingly in the main results.

Oracle Reflections
1. Three Sum example was very confusing, so it would be nice to go over it again. Also just more practice problems with determining the runtime of functions. And lastly just reiterating the formal definitions of the Big O family 2. The triple loops example and finding the runtime I got really lost in how we determine how to segment the loops into pieces and also how to calculate each individual output. Also why O(n) is the same as O(n+1) ie why the +1 is dropped from the calculation. 3. I found it hard to follow some of the math done to analyze code examples, as well as towards the beginning with the formal definitions of the big o family. 4. More practice with determining run-times would be useful as well as more practice with the Big-O family functions 5. Mathematically modeling runtime was somewhat confusing as was determining the cost of executing each statement.
GPT-reflect output
1. Three Sum example was very confusing, so it would be nice to go over it again. 2. Just more practice problems with determining the runtime of functions. 3. The triple loops example and finding the runtime I got really lost in how we determine how to segment the loops into pieces and also how to calculate each individual output. 4. I found it hard to follow some of the math done to analyze code examples, as well as towards the beginning with the formal definitions of the big o family. 5. Mathematically modeling runtime was somewhat confusing as was determining the cost of executing each statement.

Table 13: An example, the first two GPT model outputs came from the same oracle reflection, and the third GPT model sentence belongs to the second oracle reflection. If we only consider the exact matching, the true positive count would be 2, while partial F1 considers the true positive as 4, which is more realistic and relaxed.

C.2. Factuality Evaluation

In order to emphasize the difficulties associated with utilizing QA-based evaluation metrics for factuality assessment, we present an example in Table 14 that showcases the challenges encountered in entity extraction, which is a crucial component of the question generation module. Consequently, we opted not to incorporate QA-based factuality evaluation in our approach, acknowledging the limitations and complexities associated with this methodology.

Input reflections
• Combinatorics. • Understanding that the sum of two integers is still less then or equal to the product of twice the maximum of the set of integers being added was a little mind blowing since I was struggling understanding that for awhile. Using the product rule to consider abstract examples like possible injective functions. • Combinatorics seems really cool and it seems like it has a lot of real world and CS based applications. the visual explanation of the binary trees was more helpful than my reading about them so that was interesting The combinatorics was a topic that I feel relatively comfortable with already. It seems straightforward and easy to comprehend as of now. • The idea that induction has so many applications is interesting to me. • Combinatorics.
Extracted entities
Entity two: **Type:** number
Extracted Nounphrases
Nounphrases: Combinatorics,CS, sum, integers, two integers, sum of two integers , applications, trees, search trees ... Shortened for space

Table 14: Examples of the extracted entities and noisy nounphrases for a given input set of reflections.

C.3. Entailment Based Metrics Limitation

To better understand the limitations of entailment methods in our domain-specific summarization task, Table 15 presents an example of SUMMAC scores applied to a human-reference summary. The zero-shot score of 0.07 indicates poor factuality. Additionally, the sentence-level $SummaC_{conv}$ score of 0.21 indicates poor factuality value (second lowest bin), which is counter-intuitive given that our human-reference summaries are accurate as shown in the example. This highlights the constraints of entailment-based metrics and emphasizes the need to explore factuality metrics tailored to the nature of students reflections.

Input reflections
• One thing I found interesting was how many categories of machine learning there are. • What discrete variables were and how they can be classified. • None. • The idea of discrete and continuous labels was most interesting. • Supervised and unsupervised learning as well as discrete and continuous labels and how they all related to one another. • The process for both categorization and classification. How one is based on context or perceived similarity and the other is a systematic arrangement of entities. • How to categorize things as continuous or discrete. • Different categories of machine learning. • The algorithm systems for the ways the algorithms group different things on the way they identify the patterns. • The relationship between unsupervised and supervised deep learning. • I was intrigued by why discrete meant classification and how those 2 worked together was very interesting. • Supervised vs unsupervised learning.
Human-reference summary
Students enjoyed learning about the differences between supervised and unsupervised learning. Along with that, they also enjoyed learning about the different categories in Machine Learning and the different categorization and classification methods.
$SummaC_{conv}$: 0.21
$SummaC_{zs}$: 0.07

Table 15: SUMMAC scores on a human-reference summary example

D. Experimental Details

D.1. PhraseSum Setups

We first reproduced the model from prior work (Luo and Litman, 2015). This model utilizes the spaCy toolkit (Honnibal et al., 2020) to extract all noun phrases from student reflections, followed by the BERT-base model to extract the representation of these phrases. An unsupervised clustering method, KMedoid, is then employed to construct five clusters and extract their centroids as the final extractive phrases. Following Luo and Litman (2015), we named it **PhraseSum**.

D.2. Reflection Specificity Prediction

The goal of this task is to predict the specificity of student reflections. We first experimented with the prior model introduced by Magooda et al. (2022), which uses a DistillBERT model (Sanh et al., 2019) followed by an SVM to predict scores on a 4-point scale. The baseline model was trained on the publicly available CourseMIRROR (CM) corpus[18], consisting of 6,824 student reflections collected from four undergraduate classes (Chemistry (Chm), Statistics (ST), and Material Science (MSG1, MSG2)) at the end of each lecture. We retrained the model using a k-fold cross-validation setup. The QWK (Quadratic Weighted Kappa) scores improved from 0.624 to 0.689 when we trained the baseline model on our newly collected dataset.

D.3. Regular Expression for Phrase Cleaning

We have included the Python code snippets below, which demonstrate the regular expressions used in Section 7 to clean up the "number of support" for phrase outputs generated by GPT models. One example of applying the regular expressions together with some post-processing to remove the order numbers can be found in Table 16. Since automatic metrics are sensitive to the n-gram wordings, removing the predicted values inside the parenthesis can make the comparison fair.

```
my_regex = re.compile(r"->(1|1.0|2.0
|2|3.0|3|4.0|4)|(\|\|)")
text = my_regex.sub('', text)

my_regex = re.compile(r"\(rated.*\)")
text = my_regex.sub("", text)

my_regex = re.compile(r"rated␣as␣a␣\d")
text = my_regex.sub("", text)

my_regex = re.compile(r"->␣rating:␣(1|1.0|
2.0|2|3.0|3|4.0|4)")
text = my_regex.sub("", text)

my_regex = re.compile(r"-␣(1|1.0|2.0
|2|3.0|3|4.0|4)")
text = my_regex.sub("", text)

my_regex = re.compile(r"\(\d+␣(student)
?s?\)")
text = my_regex.sub("", text)

my_regex = re.compile(r"\(\d+(student)
?s?\)")
text = my_regex.sub("", text)

my_regex = re.compile(r"\(\s+␣(student)
?s?\)")
text = my_regex.sub("", text)
```

RegEx	Text
Before	1. Conditional probability examples (2) 2. Race with ties problem (1) 3. Challenge problems (1) 4. Sequences (1) 5. Disobeying conditional probability (1)
After	Conditional probability examples Race with ties problem Challenge problems Sequences Disobeying conditional probability

Table 16: Examples of phrases processed using regular expressions in summaries generated by GPT models.

[18] https://engineering.purdue.edu/coursemirror/download/reflections-quality-data/.

E. Output Examples

Extractive Summarization Table 18 and 19 shows an example of different system outputs for the extractive summarization models. *GPT-reflect* and *GPT-reflect+specificty* extracted the full reflections with the modified prompt. The latter extracted more detailed reflections with the provided specificity hints. We also observe that MatchSum is more capable of extracting long reflections that align better with human preferences.

Extractive Phrase Summarization Regarding the phrase summarization tasks, as shown in Table 17, adding the noun prompt can make the generated phrases similar to the human written version. Yet, it is still unclear how the GPT models generate the student support count for each phrase (for both *GPT-Human* and *GPT-Human + noun*).

Abstractive Summarization In the context of abstractive summaries, Table 20 provides examples of the generated summaries compared to those written by humans. Notably, models based on fine-tuned *BART-Large* (BART-Large and Bart-Large + specificity) demonstrate a tendency to cover a broader range of topics compared to the reference summaries. For instance, the generated summary includes details about concepts such as "deep copy," "shallow copy," and specific implementation and usage of the "BinaryNode." Conversely, *GPT-Human* tends to produce shorter summaries that do not delve into deeper details, aligning with our analysis indicating that *GPT-Human* summaries are generally shorter compared to the fine-tuned BART models. *GPT-one-shot* generates summaries that encompass interesting topics related to confusion topics. We hypothesize that this might be due to the random selection of examples for one-shot learning, which could lead the model to capture intriguing aspects of the lecture content.

Oracle Phrase Summary
1. Ohm's Law (4) 2. Resistors/Resistance (6) 3. Circuits (4) 4. Power (2) 5. Real world applications (3)
PhraseSum
1. the copper wire example 2. Ohm's Law 3. the resistors 4. real world incidences 5. how easy calculating resistance
GPT-Human
1. Ohm's Law and circuit analysis (8) 2. Resistance in series and parallel (4) 3. Battery voltage and electromotive force (3) 4. Applying physics concepts to real life (2) 5. Power and energy in circuits (2)
GPT-noun phrase
1. Copper wire example 2. Ohm's Law 3. Circuits 4. Batteries and resistance 5. Practice problems
GPT-Human + noun
1. Circuit analysis (4) 2. Ohm's Law (3) 3. Resistance calculation (2) 4. Battery behavior (2) 5. Power and energy (2)
GPT-noun - one-shot
1. Copper wire example 2. Ohm's Law 3. Circuits 4. Resistance calculations 5. Batteries and their characteristics

Table 17: An example of the extractive phrase summary and different model outputs.

Oracle Reflections
1. The idea of grounding was confusing for me. I feel like seeing some more examples would be helpful. 2. Problem 27.54 with the car motor problem was confusing. I did not understand what the starting motor does. Does it withdraw or add voltage? Does it act as a resistor or a battery? Otherwise, your lectures were clear and concise as always. 3. I am still a bit confused about why the voltage was negative in the first given example even though the voltage coming out of the battery is positive. 4. Current for the last top hat circuit question that was graded. 5. Re-drawing circuit diagrams with grounds seems to get complicated when multiple branches and resistors/-capacitors are involved.
LexRank
1. Some of the ideas about grounding were confusing, especially with the top hat, but I think I have the general idea that a grounded circuit has a voltage drop of 0. 2. Some of the ideas about grounding were confusing, especially with the top hat, but I think I have the general idea that a grounded circuit has a voltage drop of 0. 3. The most confusing thing for me was when we went over the 2 different cases of grounding. I just found it confusing when going over the formulas like whether to add or subtract IR. 4. I found grounding to be very confusing; what does it mean for a circuit to be grounded? 5. Conceptually, I was a little confused how you can have a ground before a resistor, making the voltage zero, and that doesn't impact the energy continuing on. Mathematically I can see why this is true, but conceptually it's a bit confusing.
GPT-reflect
1. The idea of grounding was confusing for me. I feel like seeing some more examples would be helpful. 2. I'm confused about how the spot where we place a ground affects how we measure voltage. 3. The most confusing thing for me was when we went over the 2 different cases of grounding. 4. The only thing that confuses me slightly is the sign associated with IR when going in the flow of charge vs against the flow of charge. Also, how would we determine where the charge flows? 5. Finding the emf was confusing.
GPT-reflect + specificity
1. I am still a bit confused about why the voltage was negative in the first given example even though the voltage coming out of the battery is positive. 2. Some of the ideas about grounding were confusing, especially with the top hat, but I think I have the general idea that a grounded circuit has a voltage drop of 0. 3. The conceptual questions during class were the hardest for me to understand because I kept getting the wrong. 4. It was really hard to understand the currents and why they were going in the direction they were. 5. The idea of grounding was confusing for me. I feel like seeing some more examples would be helpful.
GPT-reflect one-shot
1. The conceptual questions during class were the hardest for me to understand because I kept getting the wrong answers. 2. The idea of grounding was confusing for me. I feel like seeing some more examples would be helpful. 3. More examples of simplifying circuits would be helpful and a bit more explanation about grounding. 4. The most confusing thing for me was when we went over the 2 different cases of grounding. I just found it confusing when going over the formulas like whether to add or subtract IR. 5. The sapling question about finding emf and internal resistance has me stumped. If we could go over more of these that would be awesome.

Table 18: An example of the difference between the oracle extractive summary and the generated model outputs (1/2).

Oracle Reflections
1. The idea of grounding was confusing for me. I feel like seeing some more examples would be helpful. 2. Problem 27.54 with the car motor problem was confusing. I did not understand what the starting motor does. Does it withdraw or add voltage? Does it act as a resistor or a battery? Otherwise, your lectures were clear and concise as always. 3. I am still a bit confused about why the voltage was negative in the first given example even though the voltage coming out of the battery is positive. 4. Current for the last top hat circuit question that was graded. 5. Re-drawing circuit diagrams with grounds seems to get complicated when multiple branches and resistors/-capacitors are involved.
BERTSUM-EXT (cnndm)
1. I am still a bit confused about why the voltage was negative in the first given example even though the voltage coming out of the battery is positive. 2. Problem 27.54 with the car motor problem was confusing. I did not understand what the starting motor does. Does it withdraw or add voltage? Does it act as a resistor or a battery? Otherwise, your lectures were clear and concise as always. 3. The idea of grounding was confusing for me. I feel like seeing some more examples would be helpful. 4. Some of the ideas about grounding were confusing, especially with the top hat, but I think I have the general idea that a grounded circuit has a voltage drop of 0. 5. I found grounding to be very confusing; what does it mean for a circuit to be grounded?
BERTSUM-EXT (ft. FEWSUM AMAZON*)*
1. I found grounding to be very confusing; what does it mean for a circuit to be grounded? 2. Why it is useful to have a reference point. I am confused about what exactly delta V is. 3. The most confusing thing for me was when we went over the 2 different cases of grounding. I just found it confusing when going over the formulas like whether to add or subtract IR. 4. I'm confused about how the spot where we place a ground affects how we measure voltage. 5. The sapling question about finding emf and internal resistance has me stumped. If we could go over more of these that would be awesome.
BERTSUM-EXT (ft. REFLECTSUMM*)*
1. Conceptually, I was a little confused how you can have a ground before a resistor, making the voltage zero, and that doesn't impact the energy continuing on. Mathematically I can see why this is true, but conceptually it's a bit confusing. 2. I am still a bit confused about why the voltage was negative in the first given example even though the voltage coming out of the battery is positive. 3. Finding the emf was confusing. 4. Why it is useful to have a reference point. I am confused about what exactly delta V is. 5. Figuring out how grounding will affect the potential.
BERTSUM-EXT (ft. REFLECTSUMM*) + Specificity*
1. I am still a bit confused about why the voltage was negative in the first given example even though the voltage coming out of the battery is positive. 2. It was really hard to understand the currents and why they were going in the direction they were. 3. Current for the last top hat circuit question that was graded. 4. Some of the ideas about grounding were confusing, especially with the top hat, but I think I have the general idea that a grounded circuit has a voltage drop of 0. 5. Figuring out how grounding will affect the potential.
MatchSum
1. Some of the ideas about grounding were confusing, especially with the top hat, but I think I have the general idea that a grounded circuit has a voltage drop of 0. 2. The clicker questions about grounding. 3. Problem 27.54 with the car motor problem was confusing. I did not understand what the starting motor does. Does it withdraw or add voltage? Does it act as a resistor or a battery? Otherwise, your lectures were clear and concise as always. 4. Finding the emf was confusing. 5. I found grounding to be very confusing; what does it mean for a circuit to be grounded?

Table 19: An example of the difference between the oracle extractive summary and the generated model outputs (2/2).

Human-reference
In this lecture, students were most confused with the BinaryNode class. They also require further instruction on methods to construct and implement a binary tree. Also, students were confused on the different traversal methods.
BART-Large
In this lecture, students were confused by the BinaryNode class and BinaryTree implementation. Students were also confused by BinaryNode's implementation and how it used BinaryNode objects to execute all its functions. Students also found it confusing to understand the difference between deep copy and shallow copy. They also were confused about the code for the different traversal methods. Lastly, they were confused on the binary tree diagram.
BART-Large + specificity
In this lecture, students were confused by the BinaryNode and BinaryTree methods. Students also found the code for the different traversal methods confusing in how it used BinaryNode objects to execute all its functions. They also found it confusing to learn about the difference between deep copy and shallow copy. Students were also confused by the tree diagram and the last example shown in class.
GPT-Human
The class covered topics such as deep copy vs shallow copy, Binary Tree implementation, BinaryNode objects, and tree traversal methods. Some students found these topics confusing, while others did not have any questions.
GPT-Human + specificity
Students found the implementation of BinaryNode and BinaryTree methods confusing, especially the difference between deep copy and shallow copy. Some were lost during the last example and had trouble understanding the constructor and copying a binary tree.
GPT-oneshot
Students found binary tree implementation confusing, particularly the deep copy vs shallow copy in BinaryNode class and the BinaryTree build tree method. Some were unsure of the last example discussed. Tree traversal and terminology were interesting.

Table 20: An example of the reference summary and different abstractive model outputs.

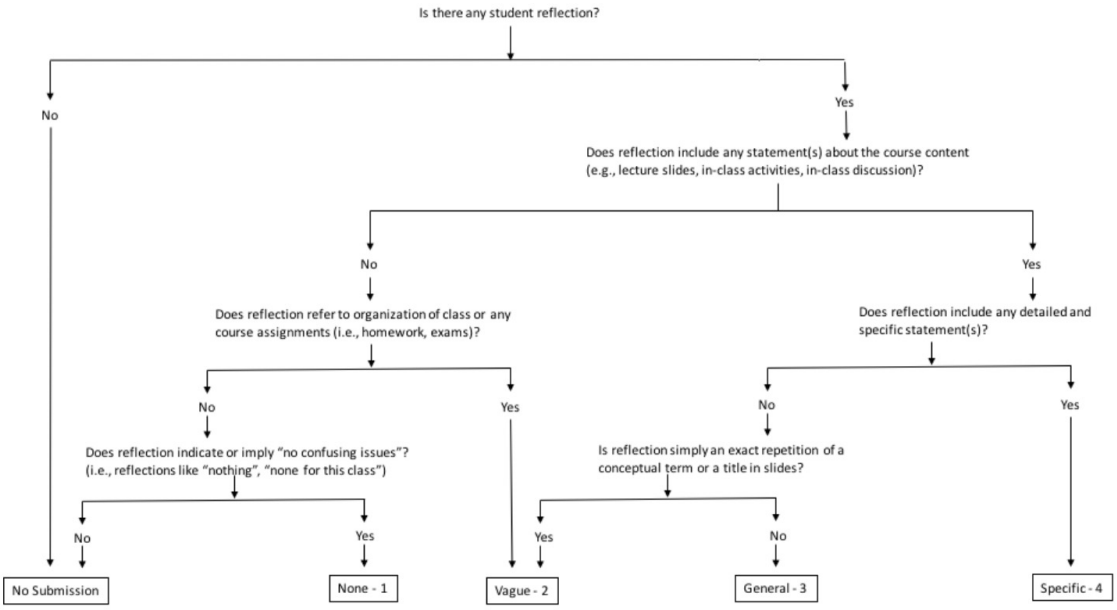


Figure 2: Human annotation guideline on the reflection specificity.

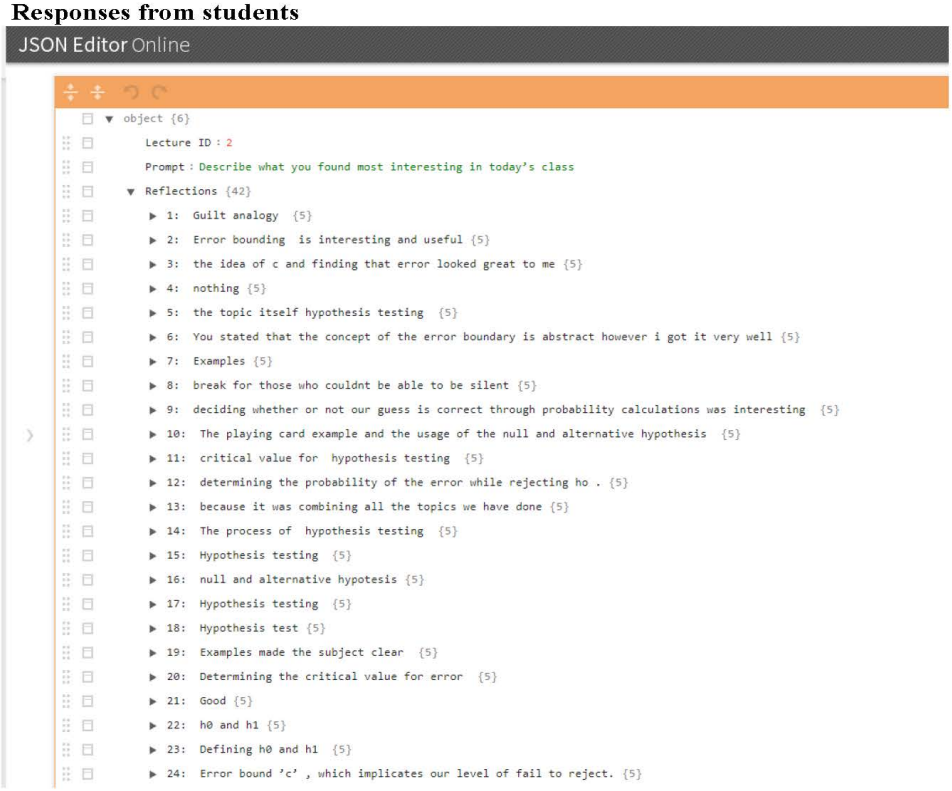
Read Me

PHRASE-BASED SUMMARIZATION ANNOTATION

In creating each summary, you should keep in mind the following scenario for its use. Imagine you are a TA for this course, what do you want to present to the instructor after reading the students' responses?

You can use any JSON editor to edit the json file provided for each set of responses. An easy to use JSON editor is http://jsoneditoronline.org/ which provides a plugin that can be installed into google chrome to easier edit JSON files.

Prompt1: "Describe what you found most interesting in today's class?"
Responses from students



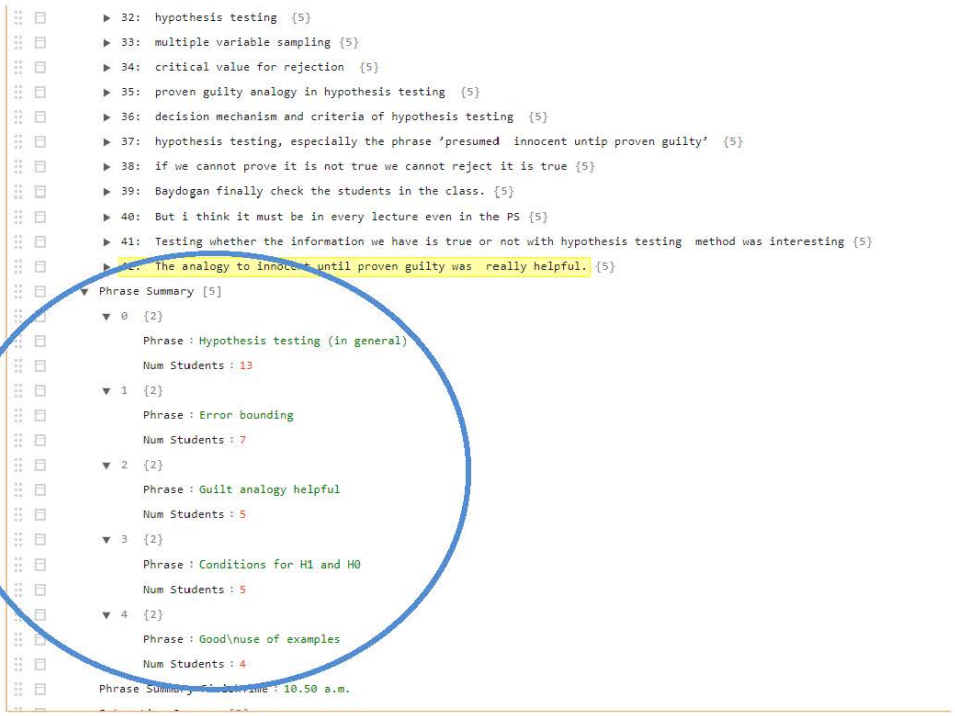
Task1: Phrase Summarization. Create a summary using 5 phrases together with how many students semantically mentioned each phrase. You can use your own phrases.

Note:

Figure 3: Human annotation guidelines 1/4.

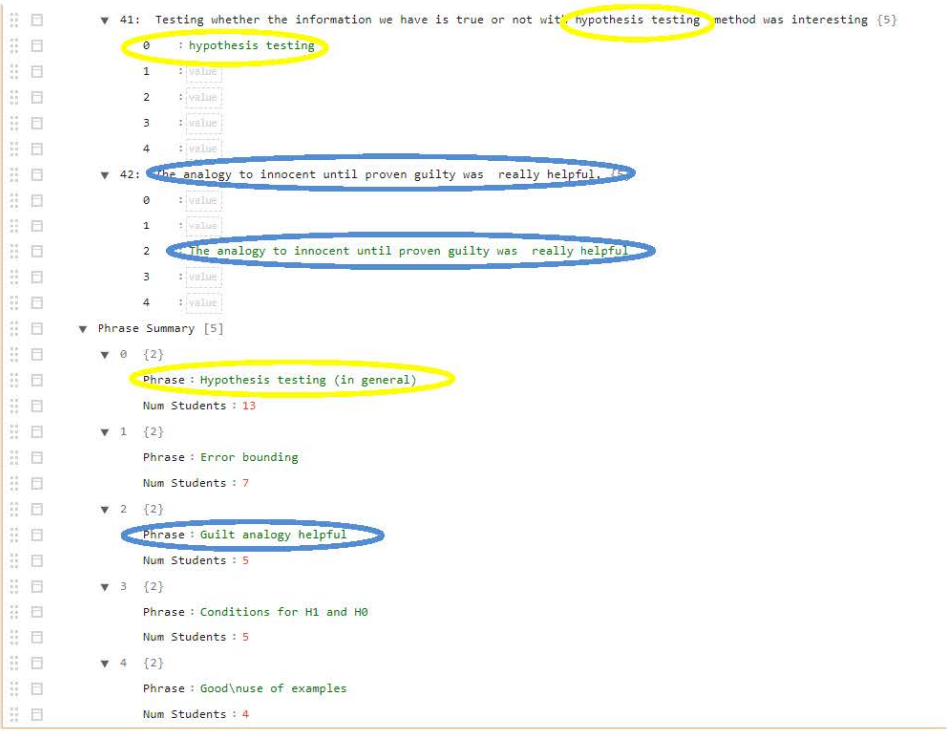
- Copy the corresponding phrases from the student responses above which are semantically similar to the summary phrases to the corresponding index below each response.
- Try to be as comprehensive as you can. extract as many phrases as you can that are relevant to the summary phrase.
- Try to have the summary phrase, and it's semantically similar extracted phrases in a sentence like shape as much as you can (i.e. extract "The analogy of innocent until proven guilty", instead of just extracting "innocent until proven guilty").
- Try to only extract phrases that are similar or very related to the summary phrase.

For example: the next figure shows the 5 phrases representing the summary.



Next we copy and paste any phrase from any response that semantically similar to any of the summary phrases below the response within the corresponding index (Next Figure)

Figure 4: Human annotation guidelines 2/4.



<u>Task2: Abstract Summarization.</u> Given the students' responses, create a short summary using your own words (~40 words) of it. The summary needs to be a coherent paragraph and should include the major points. The summary should only contain information about reflections, and avoid adding irrelevent sentences or suggestions such as " Make sure to bring this up in next class", or "Consider this for future lectures" , etc...

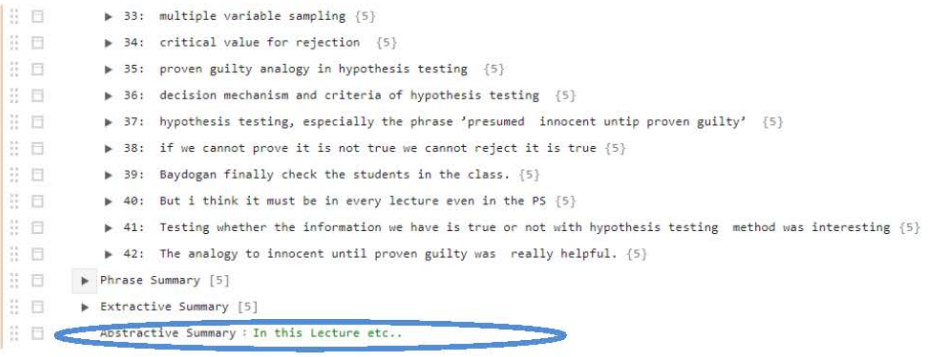
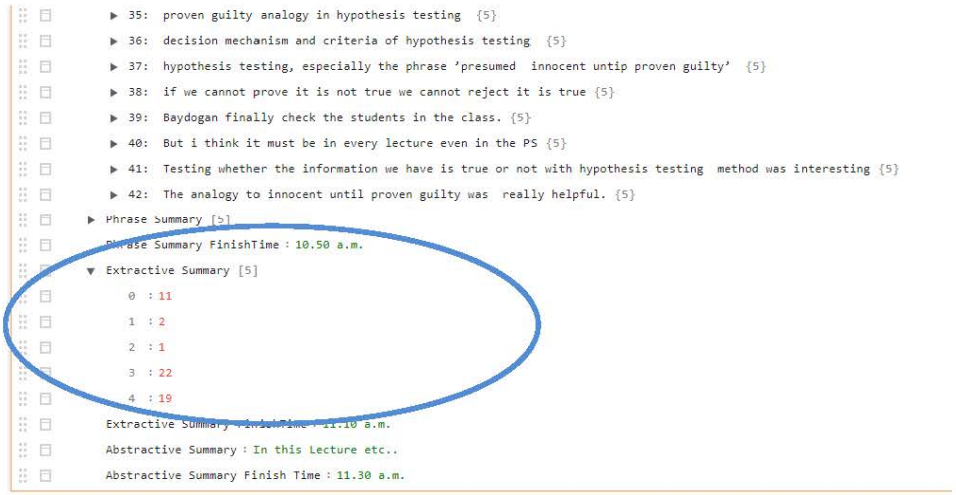


Figure 5: Human annotation guidelines 3/4.

<u>Task3: Extractive summary.</u> Select five most representative sentences in order as the summary. (Use the sentence index number.)



Final Note:
Make sure to record the starting time and the finish time for each task.

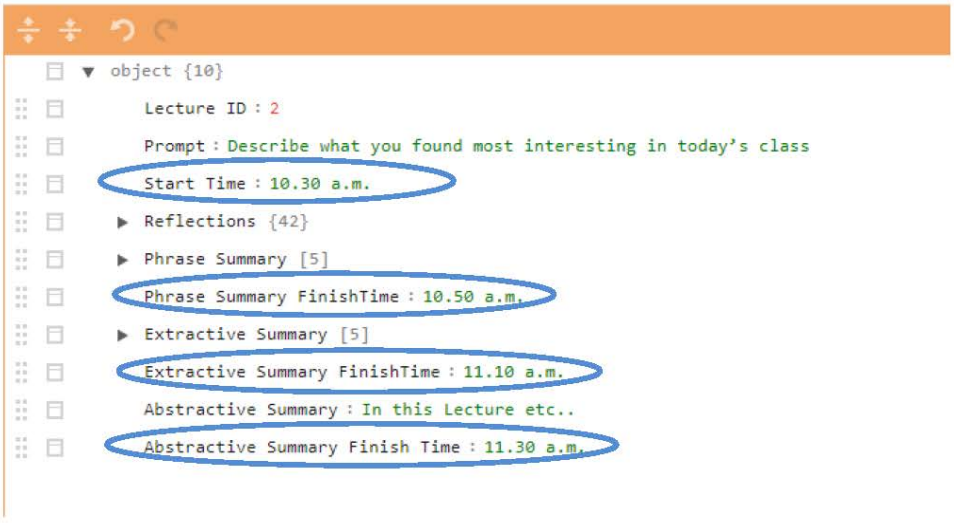


Figure 6: Human annotation guidelines 4/4.