

Human–Machine Learning for Large-Scale Qualitative Coding in Educational Measurement

Judy H. Tang (JudyTang@Westat.com),
Tom Krenzke, Jin Hui Xu, and Karen Lo
Westat

Abstract

This paper examines a human–ML framework for large-scale qualitative data coding. Using a nationally representative sample of transcript data, the study evaluates semantic embeddings and ranked recommendations through validation and user testing. Results demonstrate improved efficiency and accuracy while maintaining human expertise, oversight, and responsibility for final coding decisions.

1 Introduction

Large-scale educational assessment programs increasingly depend on complex data sources that require transforming unstructured information into standardized variables for analysis and reporting. Qualitative data coding can be labor-intensive and dependent on human judgment, particularly when information is ambiguous, inconsistent, or locally defined. Machine-learning (ML) and natural language processing (NLP) approaches offer opportunities to support these processes by generating candidate classifications to support human coding decisions and reduce the burden associated with manual coding (Gweon & Schonlau, 2024; He & Schonlau, 2020; Moreo et al., 2019; Schierholz & Schonlau, 2021).

The National Assessment of Educational Progress (NAEP) High School Transcript Study (HSTS) provides an important context for examining human–ML collaboration in large-scale qualitative coding. HSTS collects course catalog and transcript information from nationally representative samples of U.S. high school graduates. Producing these data requires transforming locally defined course information into standardized School Courses for the Exchange of Data (SCED) classifications to support

consistent analysis of student coursetaking patterns (National Center for Education Statistics [NCES], 2023). Course coding is particularly challenging because course titles, descriptions, and terminology vary across schools, districts, and states, requiring human coders to consider both textual information and educational context.

ML and NLP methods can support this process by identifying similar previously coded courses and generating candidate classifications. Semantic embeddings, for example, can represent contextual similarity among text records even when their wording differs (Reimers & Gurevych, 2019). However, ambiguous, novel, or locally defined courses may continue to require expert interpretation. A human–ML approach therefore combines the scalability of ML models with human expertise, with ML-generated recommendations supporting human coding decisions and human experts determining final classifications.

Prior research has established the potential of using ML models to automate or support qualitative classification in survey research (He & Schonlau, 2020; Moreo et al., 2019; Schierholz & Schonlau, 2021). Less attention has been given to how human–ML approaches can be integrated, evaluated, and implemented within operational educational data-processing systems. This study addresses this gap by examining a human–ML framework developed for large-scale course coding in NAEP HSTS. The framework uses semantic embeddings, similarity retrieval, and top-k recommendations to support expert human coders and is evaluated through system integration testing, output validation, counterbalanced expert-user testing, and operational implementation.

2 Human-ML Framework and Qualitative Data

2.1 Human-ML Framework

The human-ML framework combines semantic embeddings, similarity-based retrieval, top-k recommendations, and expert human coding. Course titles and descriptions are represented as semantic embeddings, allowing new course records to be compared with previously coded courses based on contextual similarity rather than exact wording (Reimers & Gurevych, 2019).

For each course requiring coding, the ML model identifies similar previously coded courses and generates ranked recommendations for SCED classifications. Coders receive the top five recommendations and associated course information to support their coding decisions. Rather than automatically assigning classifications, the human-ML framework uses ML-generated recommendations to support decision-making. Human coders review the recommendations, consider the course title and description, and determine the final SCED classification.

This human-ML approach shifts the coding process from searching across a large classification system toward evaluating a smaller set of relevant recommendations. Human expertise remains central, particularly for ambiguous, novel, or insufficiently described courses and for cases in which the ML-generated recommendations do not adequately represent the course content.

2.2 Qualitative Course Data

The human-ML framework was developed using previously coded course data from the 2019 NAEP High School Transcript Study (HSTS). HSTS collects transcript and course catalog information from a nationally representative sample of U.S. high school graduates and their schools (National Center for Education Statistics [NCES], 2023). The 2019 study included approximately 1,400 high schools and 47,000 graduates, and it collected more than 900 course catalogs containing approximately 344,000 catalog courses.

A central component of HSTS data processing involves assigning standardized School Courses for the Exchange of Data (SCED) codes to locally defined courses. SCED provides a common classification system for secondary school courses and supports consistent analysis of coursetaking

across schools and trend analysis across HSTS cycles (NCES, 2023). Course titles and descriptions vary considerably across schools and districts; similar courses may have different titles, while courses with similar titles may represent different content.

The HSTS team built the ML model utilizing previously assigned SCED codes from the HSTS 2029 catalog courses. The dataset included course titles, descriptions, and SCED codes, serving as the reference for generating similarity-based recommendations and evaluating the human-ML framework.

3 Multi-Phase Experimental Design and Evaluation

The human-ML framework was evaluated through a multi-phase experimental design that examined systems integration within the existing technical environment and coding workflow, recommendation performance, and effects on human coding performance. The evaluation included three phases: system integration testing, output validation, and counterbalanced expert-user testing. The framework was designed to support a closed-loop coding environment in which ML-generated recommendations inform human coding decisions, and finalized human-coded cases are added to the coded data available to support subsequent coding. Following these evaluations, the framework was implemented in the HSTS 2024 course-coding operation.

3.1 System Integration Testing

The first phase examined whether ML-generated recommendations could be incorporated into the existing HSTS computing environment and coding workflow. Testing focused on integrating model recommendations with the coding interface, existing databases, and established quality-control procedures.

A key component of the integration was creating a closed-loop coding environment that connected ML-generated recommendations, human coding decisions, and the coded course repository. The ML model generates recommendations based on previously coded courses; human coders review the recommendations and determine the final SCED classifications; and the system incorporates finalized coded cases into the repository to support subsequent coding. This structure allows the repository of expert-coded courses to expand as

coding progresses while maintaining human control over the final classifications.

System integration testing assessed whether this human–ML workflow could operate within the existing systems and production environment while maintaining established coding and quality-assurance procedures.

3.2 Output Validation

The second phase evaluated the performance of the ML-generated recommendations using previously coded HSTS 2019 course data. For each course, the ML model generated a ranked set of recommended SCED classifications based on similarity to previously coded courses. We compared these recommendations with the final SCED codes assigned by trained human coders.

We evaluated recommendation performance using two measures: whether the final assigned SCED code matched the first-ranked recommendation (i.e., the recommendation received the highest similarity score) and whether it appeared among the top five recommendations. The first measure assessed the precision of the model's highest-ranked recommendation, while the second assessed whether the model identified the appropriate classification within a smaller set of plausible options for human review.

Evaluating both measures reflects the human–ML framework's decision-support design. The goal was not to replace human coding with a single automated classification, but to determine whether the ML model could provide sufficiently relevant recommendations to support efficient and accurate human coding.

3.3 Counterbalanced Expert-User Testing

The third phase evaluated the effect of ML support on human coding performance using a counterbalanced experimental design. A small team of experienced HSTS coders coded 12 course groups drawn from two publicly available course catalogs. Course groups and coding conditions were distributed across coders so that each coder completed tasks both with and without ML support, while balancing differences among coders and course groups across conditions.

In the ML-supported condition, coders received the top five ML-generated recommendations, along with associated course information, to inform their coding decisions. In the non-ML condition, coders completed the same coding task

using the existing coding process without ML-generated recommendations. In both conditions, human coders retained responsibility for determining the final SCED classification.

We evaluated performance by comparing coding accuracy and coding time between the two conditions. Coding accuracy was assessed against the established SCED classifications, while coding time measured the time required to complete each course-coding decision. The evaluation also collected coder feedback on their experience using the ML-supported workflow.

4 Results

4.1 Can the Framework be Integrated into Existing Coding Operations?

System integration testing demonstrated that the human–ML framework could be incorporated into the existing HSTS coding environment without disrupting established production workflows or quality-control procedures. ML-generated recommendations were successfully integrated into the coding interface, allowing coders to review recommended SCED classifications as part of the existing coding process.

The integration also established the closed-loop coding environment described in Section 3.1. ML-generated recommendations supported human coding decisions, while finalized human-coded cases were incorporated into the coded course repository and made available to support subsequent coding. This approach allowed the expert-coded course repository to expand throughout the coding process while maintaining human responsibility for final SCED codes.

4.2 How Accurate Are the ML-Generated Recommendations?

Output validation showed that the ML model generated relevant SCED recommendations for a large proportion of courses. The first-ranked recommendation matched the final human-assigned SCED code for 72% of courses. When considering the top five recommendations, the final SCED code appeared among the recommendations for 91% of courses.

The higher top-five match rate indicates that even when the ML model did not identify the final SCED code as its first recommendation, it frequently included that code among a small set of candidate classifications available for expert

review. This result supports the framework's use of ranked recommendations as decision support rather than depending on a single ML-generated SCED course code.

4.3 Does ML Support Improve Expert Coding?

The counterbalanced expert-user testing showed that expert coders were more efficient and accurate when using ML-generated recommendations. Median coding time decreased from 35 seconds per course without ML support to 24 seconds with ML support, a 31% reduction. Coding accuracy also increased from 66% without ML support to 74% with ML support (Figure 1).

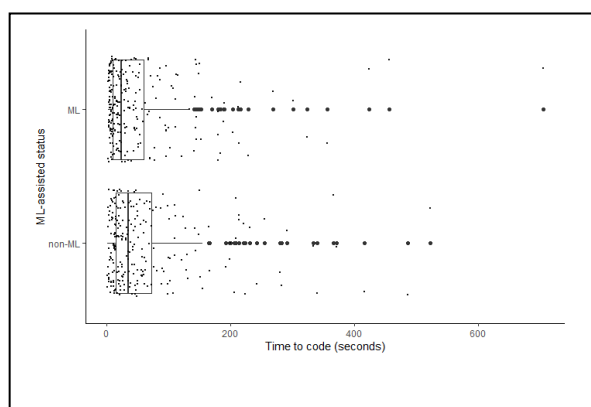


Figure 1. Expert coding performance with and without ML support.

The results indicate that ML-generated recommendations supported more efficient and accurate expert coding while maintaining human responsibility for final SCED classifications.

4.4 Can the Framework Be Implemented at Scale?

Following the evaluation, the human-ML framework was implemented in the HSTS 2024 course-coding operation. The human-ML framework supported large-scale production coding by using previously coded courses and ML-generated recommendations. Overall, the approach reduced manual coding effort by approximately 80% while maintaining established human review and quality-control procedures.

The operational implementation also extended the closed-loop coding environment to production. As coding progressed, finalized human-coded cases expanded the repository available to support

subsequent coding, allowing the framework to build on accumulated expert decisions while retaining human responsibility for final SCED classifications.

5 Discussion

The findings demonstrate the potential of a human-ML approach for supporting large-scale qualitative coding in educational measurement. Across the evaluation phases, the framework was successfully integrated into an existing coding environment, generated relevant SCED course code recommendations, improved expert coding performance, and was subsequently implemented in the HSTS 2024 production operation. These findings extend prior research on ML-assisted classification by demonstrating how ML-generated recommendations can be incorporated into an operational educational data-processing system to support expert coding decisions.

The results highlight the complementary roles of ML and human expertise. By narrowing a large classification space to a small set of relevant recommendations, the ML model reduced coders' cognitive load in searching and processing. However, the highest-ranked recommendation was not always the appropriate classification. Courses with ambiguous titles, limited descriptions, novel content, or locally specific terminology still require expert interpretation. This is especially important in large-scale educational assessment programs, where coding quality and consistency can affect the variables used for analysis and reporting.

The closed-loop coding environment further connects ML-generated recommendations with accumulated expert coding decisions. As finalized cases are incorporated into the coded course repository, they expand the body of expert-coded information available to support subsequent coding. This structure provides a mechanism for increasing the usefulness of ML support over time while maintaining human responsibility for final decisions on SCED course classifications.

More broadly, the findings suggest that the value of human-ML collaboration extends beyond reducing manual coding effort. Integrating ML-generated recommendations with expert review can support more efficient, consistent, and scalable qualitative data coding while preserving the human judgment and quality-control procedures needed for educational measurement.

6 Limitations and Future Directions

The findings should be considered within the scope of the evaluation. Expert-user testing involved a small team of experienced qualitative data coders and a limited set of course groups from two course catalogs. Although the counterbalanced design provided a structured comparison of coding with and without ML support, the evaluation was conducted within a specific HSTS course-coding context. Additional evaluation of larger coding teams and a broader range of qualitative course information would provide further evidence of the framework's performance and applicability.

Future research could also examine how the human–ML framework performs across different types of qualitative coding tasks and operational settings. Evaluating the framework in other large-scale educational survey programs and applications could help determine where ML-generated recommendations are most useful for supporting expert coding and where greater human review is needed.

7 Conclusion

This study demonstrates how a human–ML framework can support large-scale qualitative data coding by integrating ML-generated course code recommendations with expert human judgment. Applied to a large-scale qualitative course coding, the framework improved coding efficiency and accuracy, reduced manual coding effort, and was successfully implemented within an operational large-scale educational assessment program.

The findings demonstrate the value of using ML to support, rather than replace, human expertise in qualitative coding. By combining ML-generated recommendations with expert review and final coding decisions, the framework provides a practical approach for improving the efficiency, consistency, and scalability of qualitative coding while maintaining the quality and integrity of educational measurement.

Acknowledgments

The authors thank the NCME AIME-Con reviewers for their thoughtful feedback and the conference organizers for their efforts in supporting the conference and proceedings. The authors also acknowledge the HSTS project team and expert coders whose contributions supported the development and evaluation of the human–ML

framework. The views expressed in this paper are those of the authors and do not necessarily reflect the views of Westat or its clients.

References

- Gweon, H., & Schonlau, M. (2024). Automated classification for open-ended questions with BERT. *Journal of Survey Statistics and Methodology*, 12(2), 493–504.
<https://doi.org/10.1093/jssam/smad015>
- He, Z., & Schonlau, M. (2020). Automatic coding of open-ended questions into multiple classes: Whether and how to use double coded data. *Survey Research Methods*, 14(3), 267–287.
<https://doi.org/10.18148/srm/2020.v14i3.7639>
- Moreo, A., Esuli, A., & Sebastiani, F. (2019). Building automated survey coders via interactive machine learning. *International Journal of Market Research*, 61(4), 408–429.
<https://doi.org/10.1177/1470785318824244>
- National Center for Education Statistics. (2023). *2019 NAEP High School Transcript Study (HSTS) user's guide and technical report* (NCES 2023-077). U.S. Department of Education, Institute of Education Sciences.
- Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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)* (pp. 3982–3992). Association for Computational Linguistics.
<https://doi.org/10.18653/v1/D19-1410>
- Schierholz, M., & Schonlau, M. (2021). Machine learning for occupation coding—A comparison study. *Journal of Survey Statistics and Methodology*, 9(5), 1013–1034.
<https://doi.org/10.1093/jssam/smaa023>