Evaluating Deep Learning and Transformer Models on SME and GenAI Items

Joe Betts and William Muntean
National Council of State Boards of Nursing (NCSBN)
Chicago, IL

Correspondence: jbetts@ncsbn.org

Abstract

An important and time-consuming aspect of test development is the metadata coding of items within the item bank to be ready for use within the test blueprint. This study leverages deep learning, transformer models, and generative AI to streamline test development by automating metadata tagging predictions to reduce the necessary review time for subject matter experts (SME). Transformer models outperform simpler approaches and provide a direct method for reducing SME workload.

Keywords: classification models, deep learning, transformer models, Generative AI, automated item generation

1 Introduction

Developing good assessments is a time-consuming and intricate process involving numerous phases, stages, and tasks (Downing, 2006; Lane, Raymond, Haladyna, & Downing, 2016). When developing items for any assessment, subject matter experts (SMEs) are tasked with writing and reviewing items. This has traditionally been a long and expensive set of tasks. However, another aspect of the development process that is necessary but not usually identified as a high priority is related to tagging all items for their metadata content.

Metadata tagging involves classifying items by domain, task, cognitive complexity (e.g., Bloom's Taxonomy or Webb's Depth of Knowledge), or other contextual factors within a test blueprint. For this task, the SME must read each item specifically and either validate that it is representative of the task and/or domain indicated or they must provide the coding for that item. This process is labor-intensive, typically requiring SMEs to tag items or validate tags, often involving secondary reviews. However, of all the metadata content, the most important classification has to do with assigning each item to the correct test blueprint domain to ensure appropriate content to test plan blueprint alignment.

Deep learning (DL) methods and large language models (LLMs) should be useful tools in this venture as they are potentially adept at utilizing textual relationships and making predictions about content classifications. DL models, particularly those using text classification and transformer-based embeddings, can potentially reduce this workload by automating metadata tagging. This research explores how different DL and LLMs could be used to make predictions about metadata classification. Thus, building a strong model and automated pipeline could reduce SME work substantially for other work.

This study investigates DL and transformer models for natural language processing (NLP) to classify test items into test plan domains. It evaluates accessible models including Naïve Bayes (Friedman, Geiger, & Goldszmidt, 1997), XGBoost (Chen & Guestrin, 2016), deep learning models (Goodfellow, Bengio, & Courville, 2016), and some BERT family transformer models to evaluate the best approach to predicting item domain classifications. The research compares basic models to identify the

Proceedings of the Artificial Intelligence in Measurement and Education Conference (AIME-Con) – Volume 2: Works in Progress, pages 141–146

most effective approaches before exploring more complex architectures. Additionally, the best fitting model will be used against basic item generated by two different LLM models (GenAI) to see how well the model built from human curated items generalizes to GenAI.

2 Methods

2.1 Sample and GenAI Items

The study uses a subset of an item bank (N = 6,839), split into 60% training, 20% validation, and 20% testing datasets. Items were randomly selected from the group of items that had passed all statistical pretest criteria. The text data consists of item stems for multiple-choice items, labeled by eight high-level test blueprint domains (NCSBN, 2023). Table 1 provides the name of each domain and the distribution of items from each domain. The average length of the stem was 70 words (sd = 21).

For generative items using GenAI, 149 task statements (NCSBN, 2023, 2025), each tied to one of the eight domains, were used to generate 298 items across two LLMs. The prompt did not use any context about the item domain in the item generation but rather only uses the task/activity statements. Here is the prompt text:

"You are a nurse educator and clinician. Write a multiple-choice item for this entry-level nursing task: {{activity}}. The item must be a challenge to answer for an entry-level nurse. The item content should be related to this specific activity: {{activity}}. Make sure that the item content is relevant to the nursing activity. Have the item incorporate usual situations where a nurse would normally have to perform this nursing task. Provide only the item stem and the options in a json format. Do not include any other text. Do not include any explanations or rationale. Do not include any references. Do not include any other text."

Domain ID	Domain Label	Number of Items
0	Management of Care	1,202
1	Safety & Infection Control	792
2	Health Promotion & Maintenance	691
3	Psychosocial Integrity	633
4	Basic Care & Comfort	596
5	Pharmacological & Parenteral Therapies	1,202
6	Reduction of Risk Potential	721
7	Physiological Adaptation	1,002

Table 1 Domains and Number of Items

Items were created using zero-shot learning with Llama 4 Maverick (Meta AI, 2025, mixture of experts) and Claude 3.7 Sonnet (Anthropic, 2025, hybrid reasoning). While the activity statements are nested within a domain, using only the activity statements eliminates the domain context for the item generation task.

2.2 DL and Transformer Models

The analysis used two baseline models for comparing more complex DL models. These were Naïve Bayesian (NB) and XGBoost (XG). Both models used TF-IDF embeddings. NB was implemented using the Scikit-learn version 1.7.1, and XG was implemented using the XGBoost library version 3.0.4.

For DL models, a dense neural network (DNN), a convolutional neural network (CNN), a gated recurrent neural network (GRU), and a long short-term memory (LSTM) network were constructed for comparison using TensorFlow (Abadi, 2015). For embedding the text data, we used the same DL models but varied the embeddings across four different embeddings: TF-IDF, Word2Vec, GLoVe, and TensorFlow's adaptive (TFa) embeddings. Thus, for each DL model, there were four results providing 16 different conditions (four DL models x four

embeddings). This was done to evaluate the extent to which both DL model and token embeddings had any identifiable effect on results.

Data was analyzed using a cross-validation approach using training data for the model update and a validation data set for evaluating results across 100 epochs. We used an early stopping methodology with a look back of 10 epochs when the loss function of the validation data stopped decreasing. The activation function for all models was the rectified linear unit (relu), the Adam optimizer was used with learning set at 0.001, and the softmax function for output. The best model was saved as the final model and then applied to the testing data.

The CNN model used 128 filters with a kernel size of 5 and global max pooling. The GRU and LSTM models used 128 units with dropout being set to .02 and recurrent dropout set at 0.2. The DNN used 128 hidden layers. For the transformer models, we used BERT base uncased, BioBERT, and DeBERTa.

Evaluation metrics used on the classification results were accuracy, precision, recall, and F1 (Dalianis, 2018). To evaluate the similarity of text generated by the GenAI process between the two LLMs, the cosine similarity (Dalianis, 2018) was used.

3 Results

All results were based on the use of the test data and the best model trained on the training and validation data. Of the baseline models, the XGBoost (XG) outperformed the Naïve Bayes (NB) across all metrics, see Table 2.

	Metrics			
Models	***Accuracy***	***Precision***	***Recall***	***F1***
NB	0.60	0.70	0.60	0.57
XG	0.73	0.73	0.73	0.73
CNN	0.29	0.33	0.29	0.28
GRU	0.26	0.27	0.26	0.23
LSTM	0.41	0.35	0.41	0.37
DNN	0.77	0.77	0.77	0.77
BERT	0.83	0.83	0.83	0.83
BioBERT	0.83	0.83	0.83	0.83
DeBERTa	0.83	0.83	0.83	0.83

Table 2 Text Classification Metrics for Models

For the DL models, there was no appreciable difference across all of the embeddings except that the TFa was as good or better. Therefore, the results in Table 2 are reported for all of the models using the TFa embeddings. Of the DL models, only the DNN outperformed XG on all metrics with values around .77. Of interest was that the CNN and GRU models had one thing in common: high training accuracy (>90%) but poor generalization (<30% on validation and testing data). These results appeared to indicate significant overfitting. Future research should look at models with more hidden layers and a dropout regularization method to see if this improves the overfitting.

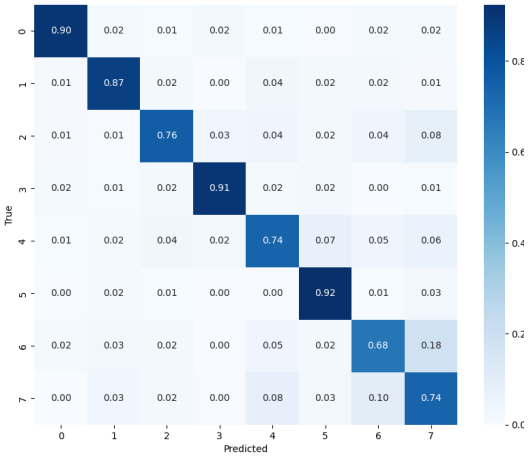

Figure 1 Cross-classification Matrix of True Domain Classification and Predicted Domain from BERT-uncased Model

The BERT family of models showed the best results across all metrics; however, there was not much separation between the models. These models accurately classified around 83% of the items. As the models were so similar, the BERT uncased model results were used for the prediction of the GenAI items.

Results of the cross-classification accuracy are provided in Figure 1 and suggested good predictions for most domains. Results were then dichotomized into correct classification = 1 and incorrect = 0. This resulted in ROC AUC = .81 and Youden's J at or above .83 suggesting an optimal trade-off in classification error (Pepe, 2004).

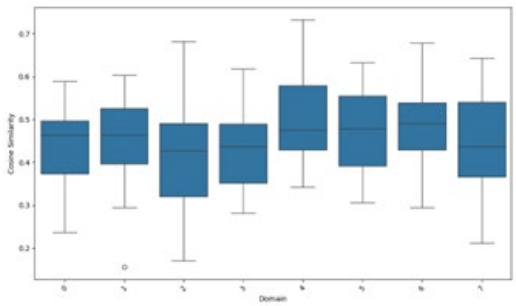

Figure 3 Box Plot of Cosine Similarity across Text Blueprint Domains

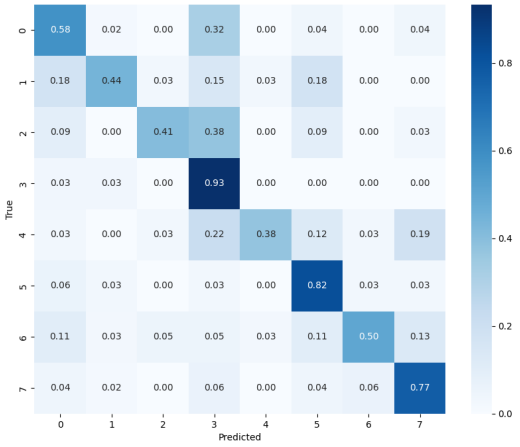

Figure 2 Cross-classification Matrix of True Domain Classification and Predicted Domain on the GenAI items from BERT-uncased Model

Item generation (GenAI) results of the 298 LLM-generated items had an average cosine similarity index between paired task statements between the models of .44 (sd = .11), see Figure 2 for box-and-whisker plot for the values by domain index. Using the best BERT uncased model results suggested a reduction in classification accuracy to around 61% which was significantly lower when compared to the classification using human generated items. Figure 3 provides the cross-classification matrix of 'true' values which was the domain for which the task statement was subsumed and the 'predicted' values from the best performing BERT model.

4 Discussion

This research investigated the extent to which different text-based models could be used for classifying assessment items into content domains. The results were varied with the baseline models having 60–70% accuracy. Of the DL models, the DNN showed the best results with about 77% accuracy. There was no clear difference between text embedding approaches across the DL models. Suggesting that, at least for this current task, any of the embedding

methods would be equivalently useful. However, this might not generalize to all assessment program content or scope. It is recommended to investigate which embeddings might be best for each program while researching and developing classification models for one's own data.

The BERT family had the best results with all metrics greater than 80%. However, the results dropped off when the best BERT model was applied to the GenAI items to around 60%. This will certainly be different for each program and for different context engineering approaches to generate items. This research used a very simple prompt with minimal context for generating the items. Future research could evaluate results across various prompts and context engineering strategies to help identify the best ones to use for the programs' distinct needs.

The utility of these results is that it suggests the potential to reduce the work of SMEs by up to 80% when tasked with coding newly written items. Additionally, this process could help with reviewing previously coded items to support on-going quality control of metadata. This type of classification accuracy has the potential to significantly reduce resource utilization on metadata coding for SMEs to focus on content development and reviews that utilize their unique expertise and domain understanding.

This research found that a cut-off of a probability of correct classification of .83 was a reasonable value to balance errors. We would encourage practitioners to utilize their own results to set the relative errors they would be willing to accept. Additionally, to ensure model validation, the SMEs should also systematically review a small percentage of the items in the neighborhood above the threshold. This way, the original model is being continually evaluated in case the model begins drifting.

Future research could expand this approach by evaluating more complex DL models. For the CNN models, applications of dropout regularization could reduce the overfitting.

Combining DL models into more complex models using the strengths of the different methods could be evaluated. Additionally, both the smaller transformer models and extending to proprietary models available like Grok, OpenAI, etc. could be promising. Finally, extension to other metadata would be useful to see if the results generalize to other important categories.

Overall, these results are encouraging. The high rate of classification accuracy has the potential to automate a time consuming and resource intensive aspect of item development. With the automation of these tasks, SMEs can focus on more relevant work to support a program's item development needs.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., et al. (2015). *TensorFlow: Large-scale machine learning on heterogeneous systems*. Software available from https://www.tensorflow.org/

Anthropic. (2025). Claude 3.7 Sonnet [large language model]. Anthropic PBC. https://www.anthropic.com/news/claude-3-7-sonnet

Chen, T., & Guestrin, C. (2016). *XGBoost: A scalable tree boosting system*. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785–794). ACM. https://doi.org/10.1145/2939672.2939785

Dalianis, H. (2018). *Clinical Text Mining: Secondary Use of Electronic Patient Records*. Springer Open.

Devlin, J., Chang, M-W, Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint: arxiv.org/abs/1810.04805

Downing, S. M. (2006). Twelve steps for effective test development. In S. M. Downing & T. M.

Haladyna (Eds.), *Handbook of test development* (pp. 3-25). Lawrence Erlbaum Associates.

Friedman, N., Geiger, D., & Goldszmidt, M. (1997). Bayesian network classifiers. *Machine Learning, 29*(1–3), 131–163. https://doi.org/10.1023/A:1007465528199

Lane, S., Raymond, M. R., Haladyna, T. M., & Downing, S. M. (2016). Test development process. In S. Lane, M. R. Raymond, & T. M. Haladyna (Eds.), *Handbook of test development* (2nd ed., pp. 3-18). Routledge.

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., & Kang, J. (2020). BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics, 36*(4), 1234-1240.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

He, P., Liu, X., Gao, J., & Chen, W. (2020). DeBERTa: Decoding-enhanced BERT with Disentangled Attention. arXiv preprint: https://arxiv.org/abs/2006.03654

Meta AI. (2025). LLaMA 4 Maverick (17B-128E) [Large language model]. Meta Platforms, Inc. https://huggingface.co/meta-llama/Llama-4-Maverick-17B-128E-Instruct

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). *Efficient estimation of word representations in vector space*. arXiv preprint arXiv:1301.3781. https://arxiv.org/abs/1301.3781

National Council of State Boards of Nursing (NCSBN). (2023). *Next Generation NCLEX: NCLEX-RN Test Plan*. National Council of State Boards of Nursing. Chicago, IL.

National Council of State Boards of Nursing (NCSBN). (2025). *2024 RN practice analysis: Linking NCLEX-RN examination to practice*. National Council of State Boards of Nursing. Chicago, IL.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12(Oct), 2825–2830. https://www.jmlr.org/papers/v12/pedregosa11a.html

Pennington, J., Socher, R., & Manning, C. D. (2014). *GloVe: Global Vectors for Word Representation*. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 1532–1543). Association for Computational Linguistics. https://aclanthology.org/D14-1162/

Pepe, M. S. (2004). *The statistical evaluation of medical tests for classification and prediction*. Oxford University Press.

Sammut, C., & Webb, G. I. (Eds.). (2011). TF–IDF. In Encyclopedia of Machine Learning (pp. 986–987). Springer. https://doi.org/10.1007/978-0-387-30164-8_832

Youden, W. J. (1950). Index for rating diagnostic tests. *Cancer*, *3*(1), 32–35. https://doi.org/10.1002/1097