Comparison of Machine Learning Methods for Multi-label Classification of Nursing Education and Licensure Exam Questions

John T. Langton **Krishna Srihasam** **Junlin Jiang**
Wolters Kluwer Health, 230 3rd Avenue, Waltham, MA 02451
{john.langto, krishna.srihasam, junlin.jiang} @wolterskluwer.com

Abstract

In this paper, we evaluate several machine learning methods for multi-label classification of text questions. Every nursing student in the United States must pass the National Council Licensure Examination (NCLEX) to begin professional practice. NCLEX defines a number of competencies on which students are evaluated. By labeling test questions with NCLEX competencies, we can score students according to their performance in each competency. This information helps instructors measure how prepared students are for the NCLEX, as well as which competencies they may need help with. A key challenge is that questions may be related to more than one competency. Labeling questions with NCLEX competencies, therefore, equates to a multi-label, text classification problem where each competency is a label. Here we present an evaluation of several methods to support this use case along with a proposed approach. While our work is grounded in the nursing education domain, the methods described here can be used for any multi-label, text classification use case.

1 Introduction

All nurses within the United States must pass the National Council Licensure Examination (NCLEX®) to begin professional practice. A nursing curriculum will typically cover a wide range of topics related to the theory and practice of nursing. However, the NCLEX measures students against a specific set of competencies comprising the activities that entry-level nurses are most commonly expected to perform. These activities are identified by the National Council of State Boards of Nursing (NCSBN) through analysis of nursing practice.

Figure 1 shows a subset of NCLEX competencies called "activity statements" with descriptions. Activity statements are grouped into primary topics and sub-topics, as shown in the image. Nursing education content may be related to one or more competency; they are not mutually exclusive.

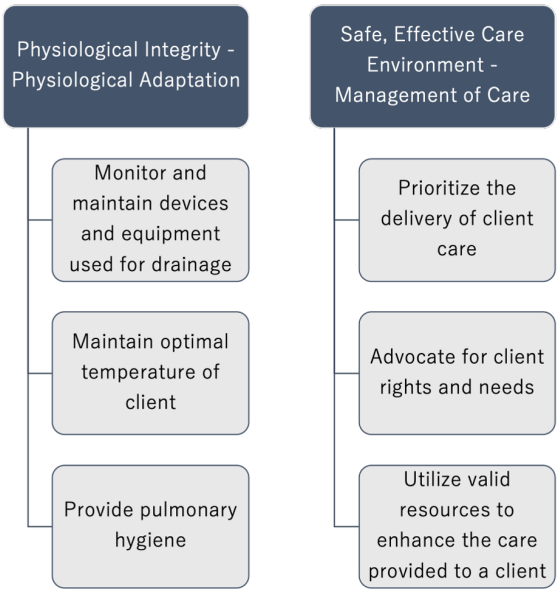


Figure 1: Sample of NCLEX competencies or activity statements.

Passage of the NCLEX has significance not only for students but also for learning institutions. Nursing school accreditation is partially based on how well their student body performs on the NCLEX. If their performance drops below a certain threshold for too many consecutive years, the school risks losing its accreditation. It is, therefore, paramount for instructors to gauge student preparedness for the NCLEX and course correct where necessary. One way to do this is by repeatedly testing students with simulated exams. This approach may reveal that gaps exist, however, it does not necessarily identify what competencies are deficient or what content may address those deficiencies. By labeling both questions and educational content with the competencies that they relate to, instructors can

Proceedings of the 3rd Clinical Natural Language Processing Workshop, pages 85–93
November 19, 2020. ©2020 Association for Computational Linguistics

more precisely identify where students are struggling and what content may help with remediation. This approach enables coursework to be tailored for individual students based on their performance in a manner that maximizes likelihood of passing the NCLEX. For instance, if a student incorrectly answers questions related to "Provide pulmonary hygiene" (activity statement shown in Figure 1), the instructor may assign the student additional content (e.g., simulations and practice problems) related to that competency. This general approach is called formative testing.

PrepU is a Wolters Kluwer product for nursing education that features several types of content including books, simulations, videos, audio, and quizzes. To support formative testing in PrepU, quiz questions are tagged with the NCLEX competency related to them. When students take quizzes, their scores can be aggregated according to NCLEX competency. Figure 2 shows an example of the interface displaying this information. The image shows the student achieves a score of 66.7% for the competency "Prioritize the delivery of client care". This is calculated based on the student answering 4 out of 6 questions correctly that were labeled with that competency. There is also a tab that shows class performance so that instructors can see if there is a pattern of multiple students struggling with a particular competency. Instructors can use this information to make changes to the curriculum to address problem areas. Corrective actions may include assigning students additional content or practice materials related to the competencies they are struggling with.

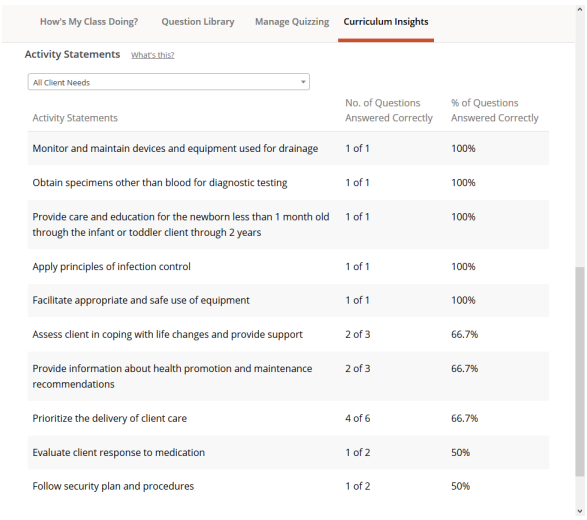


Figure 2: PrepU screenshot showing quiz results broken down according to NCLEX competencies.

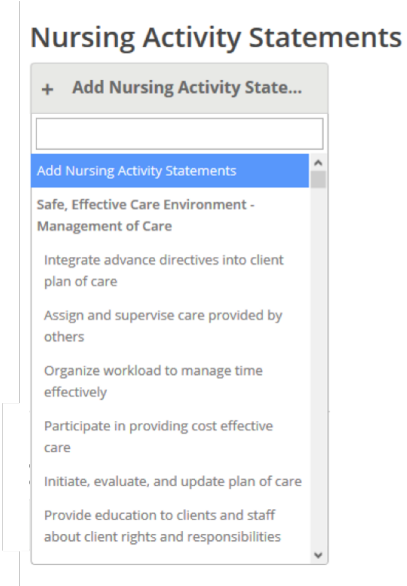


Figure 3: Editorial platform where editors manually tag questions with associated NCLEX competencies.

To aggregate scores according to NCLEX competencies as shown in Figure 2, each question needs to be labeled according to which competencies it relates to. Prior to our work, editors would manually label questions using the editorial platform shown in Figure 3. A drop-down menu shows a selection of NCLEX competencies. The editor must scroll through this list, identify which are appropriate, and select them to add them to the question. This process was costly and time-consuming. One challenge is that each question can belong to more than one competency. Further, different editors may have differing opinions as to which competencies a question relates to. Reconciling these differences and maintaining consistency across editors and content is a huge challenge.

To streamline the labeling of nursing education questions, we integrated a machine learning model for automated tagging into the current workflow. As editors review each question, the model makes suggestions about which NCLEX competencies are related to that question. Rather than scrolling through a long list of options, editors can rapidly click to accept or reject suggestions (though they still have the ability to scroll through all possibilities if they believe none of the suggestions are applicable). This approach has greatly streamlined the process of labeling questions and added additional consistency in the application of labels. The following sections detail the data involved, the modeling techniques evaluated, and the chosen solution.

2 The Data

In this paper, we focus on NCLEX competencies related to what are called "activity statements". Activity statements are presented in a hierarchical structure with two levels as shown in Figure 1. We consider only leaf nodes to simplify the problem. Given this consideration, there are 138 activity statements or labels in total.

41125 questions were manually labeled with one or more of the 138 possible activity statements related to them. This data was used for both training and testing of our machine learning models. The distribution of questions across activity statements was non-uniform and presented a class imbalance challenge. The majority of activity statements were assigned to 5 or fewer questions. However, there was a small set of activity statements that were commonly used, and two that were associated with nearly 3000 questions. The distribution of questions to activity statements is shown in Figure 4. Each bar corresponds to one of the 138 activity statements and its height represents the number of questions assigned to it.

Less than 100 questions that were assigned more than one activity statement label. However, there was a desire to accommodate multiple activity statements per question for future labeling efforts. Therefore, we maintained an approach using multi-label classification techniques.

3 Related Research

The task of tagging questions with relevant activity statements can be considered a multi-label document classification task where each question is a document. There are several well-known methods for this type of task. Many of them represent a document as a vector of numbers. We can use similarity and/or distance metrics between document vectors to perform several operations such as clustering and classification. A key set of decisions is how to represent documents as vectors, and what distance metrics to use for comparing them. The following sections describe a number of approaches for document vectorization as well as methods for multi-label classification.

3.1 Text Vectorization and Classification

Bag of words approaches for document vectorization are quite common and have been used with a number of different algorithms (Mccallum and Nigam, 2001). These approaches use word frequency to determine vector representations for documents and may employ a number of feature selection and normalization techniques (Xu et al., 2009). One dominant technique is called *Term Frequency – Inverse Document Frequency* (TF-IDF).

While bag of words methods have proven quite effective, they suffer a number of weaknesses. When paired with algorithms such as naïve Bayes classifiers, there is no consideration of word order, proximity, or co-occurrence within a document. This can be somewhat mitigated using n-gram techniques (i.e. considering n consecutive words as one element in document vectors). Synonyms can also confound bag of word approaches since two or more words may appear as unique elements in a document vector despite being semantically equivalent. For instance, "water" and "H2O" may show up as distinct vocabulary terms in a TF-IDF vector. When computing the cosine similarity between the vector for a document that discusses "water" and one that discusses "H2O", the result would inaccurately indicate they were dissimilar.

Word embeddings using neural networks are a more recent and popular method for vectorizing text (Kim, 2014). Long Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) are recurrent neural network (RNN) models that leverage connections between adjacent nodes in a single layer to better address word order and context. Huang, Xu, and Yu (Huang et al., 2015) compare several ensembles of bidirectional LSTMs and Conditional Random Fields (CRF) for sentence classification. Neural network models have specifically been used for multi-label document classification (Baumel et al., 2017) (Lenc and Král, 2017).

One of the most recent advances in natural language processing with neural networks is the use of pretrained, deep transformer models such as BERT (Devlin et al., 2018). BERT has outperformed many competing methods in standard language understanding tasks and has been used specifically for document classification (Adhikari et al., 2019). There is a great deal of research combining these different approaches for multiple use cases.

3.2 Multi-label Classification

Multi-label classification refers to a classification problem where each item being classified can belong to more than one class (or label) at the same time. This contrasts with standard classification

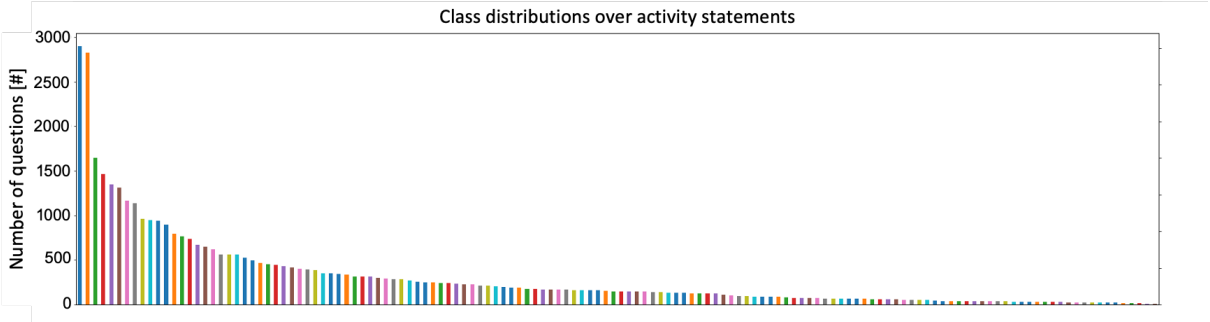


Figure 4: Sample distributions of number of questions per NCLEX activity statement.

where each item is assigned to only one class. A trivial example would be classifying geometric shapes where a square could be both a square and a rectangle.

There are a few standard techniques for dealing with multi-label classification. Many transform the problem into a standard classification task. One approach is to train a binary classifier for every label independently. Each classifier is then executed on the same input to predict whether its associated label should be applied (Read et al., 2015). In this scenario, each classifier only has the knowledge of one label and only makes predictions for membership or non-membership in that label group or class. This strategy is similar to "one-versus-rest" approaches, however, it often employs techniques more analogous to one class classification or anomaly detection. An extension to this method is to chain multiple binary classifiers together in a sequence. The predictions from one classifier is passed as a feature to the next classifier until a final set of predictions is output. Probabilistic methods can be used to optimize the order of classifiers.

Another common approach for multi-label classification is to take the power set of label permutations and treat each as an independent class. This approach transforms the problem into a standard multi-class classification task. Newton et. al. compare a number of methods for such problem transformations (SpolaôR et al., 2013). For instance, we can transform a set of 3 labels, (A, B, C), into a power set of classes: $\{(A), (B)(C), (A, B), (A, C), (B, C)\}$.

4 Evaluating Vectorization Methods and Similarity Metrics for Clustering and Classification

We began our analysis by evaluating how different vectorization techniques and similarity metrics perform at differentiating questions related to one label (i.e., activity statement) from another. The ability to differentiate questions in this manner directly affects the performance of classification and clustering algorithms. The results helped establish a baseline of how much overlap there was between questions in different label groups. It also informed decisions on which vectorization methods and similarity metrics to use with what algorithms for evaluation.

To perform this analysis, we leveraged techniques often used in clustering. The nursing education questions were grouped into clusters based on the activity statements they were associated with. This resulted in 138 clusters, one for each of the activity statements. Questions associated with more than one activity statement were included in the groups for each. We experimented with several vectorization methods (techniques for transforming the questions into numeric vectors) as well as similarity metrics for comparing vectors. We converged on using cosine similarity to compare vectors because of its ability to deal with both sparse and dense vectors when normalized. The vectorizations evaluated included the following:

- term frequency – inverse document frequency (TF-IDF)
- word embeddings pretrained on google news [(Mikolov et al., 2013)]
- word embeddings pretrained on PubMed [(Pyysalo et al., 2013)]
- word embeddings pretrained on PubMed and updated on text content from Wolters Kluwer nursing education

For each vectorization method, we computed the silhouette score across our manually constructed

clusters. The silhouette score measures the similarity of questions within a cluster (cohesion) versus the dissimilarity of questions in one cluster as compared to those in other clusters (separation). Higher silhouette scores indicate better cohesion within clusters and separation between clusters. In clustering, this measure can help inform the number of clusters to use. For our analysis, we were more interested in what vectorization methods achieved better separation of questions assigned to different NCLEX labels. The vectorizations achieving the best silhouette scores could be expected to perform better in classification tasks. We therefore controlled the number of clusters to the number of activity statements, i.e. 138.

Table 1 shows the different vectorization methods evaluated along with their respective silhouette scores. We also include metrics based on the cohesion component of the silhouette score. Specifically, the binary relevance scores measure the distance between question vectors that are all tagged with the same label. The table reports the mean, minimum, maximum, and standard deviation of binary relevance scores across all 138 label clusters.

Figure 5 shows the silhouette scores for every pair of activity statement clusters using TF-IDF vectorization of questions. TF-IDF resulted in the highest silhouette score of -.02. However, the scores for all vectorization methods were relatively low. This result indicated two things 1) there is a great degree of similarity between questions assigned to different activity statement labels, and 2) no vectorization method performed much better than the others. This result indicated that algorithms may need further grouping and sampling of questions to better differentiate them during classification.

We hypothesized that ignoring the current labels and clustering questions may achieve better separation for classification algorithms. To evaluate this hypothesis, we performed a standard clustering of questions using the various vectorization methods. Normally we would optimize the number of clusters based on the silhouette score or other related metrics. However, in the interest of time, we used a fixed number of 512 clusters. This number was estimated from the number of questions and their distribution across activity statements. The resulting silhouette scores improved by .02 on average but did not reflect a significant change. On average, each cluster contained questions from five different

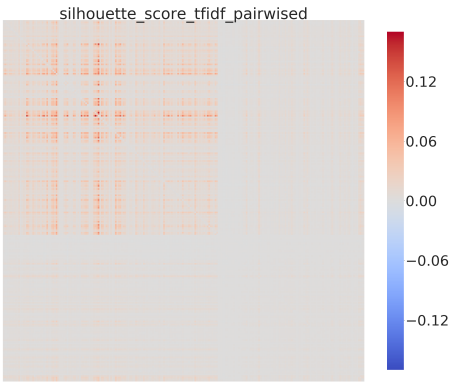


Figure 5: Distribution of pairwise silhouette scores across activity statements.

activity statement label groups. This result motivated some of the modeling experiments described in the following section.

5 Modeling

We use micro-averaged area under the receiver operating characteristic (AUC-ROC) to compare multiple algorithms for the described use case. This metric is able to address class imbalance and is used canonically for benchmarking models (Harutyunyan et al., 2019). Note that the metrics reported here only reference the AUC score of first label predicted. In production, the top five labels with the highest confidence values are shown to users and results in an accuracy of 95% in predicting all relevant labels. This is discussed further in Section 5.6. Nonetheless, the initial AUC score of the first prediction was a good benchmark for comparing models. The following sections provide details on each algorithm evaluated.

5.1 One versus Rest Support Vector Machines (SVM) with TF-IDF Vectorization

The first model employed a variant of TF-IDF vectorization referred to as Term Frequency – Inverse Label Frequency (TF-ILF). The primary difference is in what is regarded as a "document". Instead of individual questions being treated as a document, questions are grouped according to their labels and then the group is treated as a document.

Vectorization Method	Silhouette Score	Mean Binary Relevance	Max Binary Relevance	Min Binary Relevance	Std Dev Binary Relevance
TF-IDF	-0.02	0.0	0.01	-0.003	0.0001
Google	-0.21	-0.02	0.09	-0.14	0.04
PMC	-0.22	-0.02	0.17	-0,13	0.05
PrepU	-0.91	-0.07	0.3	-0.27	0.08

Table 1: Silhouette scores for different vectorization Methods

This document specification results in a slight difference in how document vectors are normalized. The vocabulary for the vectorization included the use of bi-grams and tri-grams (i.e., 2 and 3 word sequences). After eliminating stop words (e.g., "a", "and", "the"), stemming, and performing synonym replacement, the total vocabulary was constrained to 30,000 features. Specifically, we kept only the 30,000 features with the highest TF-ILF values.

All questions were first vectorized. Each vector was then labeled according to the manually assigned labels using the LabelEncoder python class from the SciKit Learn package (Pedregosa et al., 2011). To address the multi-label issue (each question could have more than one activity statement label), we employed a one-versus-rest approach using support vector machines. A binary classifier was trained for each activity statement label with SciKit Learn's LinearSVC algorithm. This approach resulted in 138 models. The hyperparameters for the algorithm included an L2 penalty and ran with 1000 maximum iterations. Models were evaluated using cross-validation with the CalibratedClassifierCV python class. Cross-validation provides a more robust evaluation and can reveal variability between multiple executions of the algorithm.

Given a question as input, each classifier would predict whether a single activity statement label should be assigned to the question, without regard to any other labels. We computed confidence thresholds for each classifier as to whether to accept its prediction or not. Thresholds were established using evaluation metrics such as recall and precision. Questions were then fed into each classifier and any label predictions that met the required thresholds were assigned. In this manner, questions could be assigned more than one label provided more than one model prediction met the required threshold.

The micro-averaged AUC-ROC of the SVM model was .968.

5.2 Convolutional Neural Network

A convolutional neural network (CNN) model was trained using a Keras tokenizer and word embeddings pretrained on articles from PubMed [(Pyysalo et al., 2013)]. LabelEncoder was used again to encode question labels. The model used a softmax activation function and categorical cross-entropy for the loss function. The output of the model was structured as per-label probabilities between 0 and 1. The softmax output enabled more than one label to have a non-zero probability for a given question input therefore addressed the multi-label problem. Details of the network architecture are shown in Figure 6.

The micro-averaged AUC-ROC of the CNN model was .972.

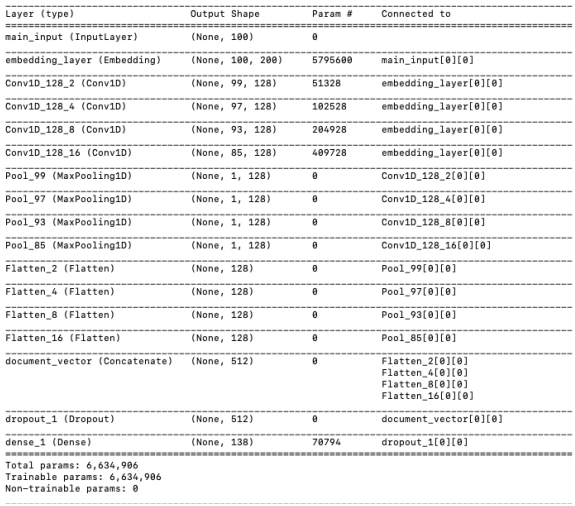
5.3 Bidirectional LSTM

A bidirectional LSTM was trained using many of the same parameters as the CNN including a softmax activation function, categorical cross-entropy loss function, and word embeddings pretrained on PubMed. The important difference in this neural network is that layer nodes were connected in a sequential manner, both forward and backwards. Attention was also used to bias more important weights in the network architecture. Details of the network architecture are shown in Figure 7.

The micro-averaged AUC-ROC of the bidirectional LSTM model was .940.

5.4 Random Forrest Ensemble of SVM, CNN, and LSTM Models

All models had similar AUC metrics. We hypothesized that different models may perform well on different subsets of labels. If this were true, it would be possible to combine the models in an ensemble to increase overall performance across all labels. We trained an ensemble classifier to evaluate this hypothesis.



Layer (type)	Output Shape	Param #	Connected to
main_input (InputLayer)	(None, 100)	0	
embedding_layer (Embedding)	(None, 100, 200)	5795600	main_input[0][0]
Conv1D_128_2 (Conv1D)	(None, 99, 128)	51328	embedding_layer[0][0]
Conv1D_128_4 (Conv1D)	(None, 97, 128)	102528	embedding_layer[0][0]
Conv1D_128_8 (Conv1D)	(None, 93, 128)	204928	embedding_layer[0][0]
Conv1D_128_16 (Conv1D)	(None, 85, 128)	409728	embedding_layer[0][0]
Pool_99 (MaxPooling1D)	(None, 1, 128)	0	Conv1D_128_2[0][0]
Pool_97 (MaxPooling1D)	(None, 1, 128)	0	Conv1D_128_4[0][0]
Pool_93 (MaxPooling1D)	(None, 1, 128)	0	Conv1D_128_8[0][0]
Pool_85 (MaxPooling1D)	(None, 1, 128)	0	Conv1D_128_16[0][0]
Flatten_2 (Flatten)	(None, 128)	0	Pool_99[0][0]
Flatten_4 (Flatten)	(None, 128)	0	Pool_97[0][0]
Flatten_8 (Flatten)	(None, 128)	0	Pool_93[0][0]
Flatten_16 (Flatten)	(None, 128)	0	Pool_85[0][0]
document_vector (Concatenate)	(None, 512)	0	Flatten_2[0][0] Flatten_4[0][0] Flatten_8[0][0] Flatten_16[0][0]
dropout_1 (Dropout)	(None, 512)	0	document_vector[0][0]
dense_1 (Dense)	(None, 138)	70794	dropout_1[0][0]

Total params: 6,634,906
Trainable params: 6,634,906
Non-trainable params: 0

Figure 6: Architecture of convolutional neural network for multi-label text classification.

Layer (type)	Output Shape	Param #
main_input (InputLayer)	(None, 100)	0
embedding_layer (Embedding)	(None, 100, 200)	7284800
LSTM_128 (Bidirectional)	(None, 100, 256)	336896
attention_weighted_average_1	[(None, 256), (None, 100)	256
dropout_1 (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 138)	35466

Total params: 7,657,418
Trainable params: 7,657,418
Non-trainable params: 0

Figure 7: Architecture of LSTM recurrent neural network for multi-label text classification.

A random forest model was trained on the outputs of the previously described models to weight the predictions of each and make a final prediction. Questions were first vectorized and input to each component model (i.e., the SVMs, CNN, and LSTM). The output probabilities of each model was then fed into the random forest. Specifically, the inputs of the random forest were 414 values between 0 and 1 consisting of:

- a probability from each of 138 binary SVMs
- a probability for each of 138 output nodes of the CNN
- a probability for each of 138 output nodes of the LSTM

The output of the random forest was a binary vector of 138 elements. Each element corresponded to an activity statement label. A value of 1 indicated the input question should have that label and a value of 0 indicated that it should not.

Modeling Method	AUC-ROC (micro-avg)
TF-ILF+SVM	0.968
CNN	0.972
LSTM	0.940
Ensemble	0.937

Table 2: AUC of different methods

The micro-averaged AUC-ROC of the random forest ensemble combining the output of the other models was .937.

5.5 Model Comparison and Discussion

Table 2 shows the micro-averaged AUC-ROC of the models evaluated. The best performing model was the CNN. However, none of the algorithms performed dramatically different from one another. We believe that several confounding factors in the data were equally challenging for the various methods.

Class imbalance likely complicated classification attempts and may also indicate other issues with the manual labeling process. Editors, pressed for time, may choose labels that are higher in the drop-down list of the editorial platform. They may also choose labels that are less precise but more general and, therefore, likely to be acceptable. These behaviors could explain why a small set of activity statements labels were associated with thousands of questions whereas the rest of the labels were only associated with a few questions each.

We also found that editors sometimes disagree about question labels. To address this issue, there is a manual process for label reconciliation. Senior editors can be consulted to make final decisions where necessary. Editors also pointed out that questions could be assigned far more activity statements than is currently the case. To optimize the adaptive quizzing experience for users, editors limit labeling to one or two labels that best fit the question.

5.6 Final Model Evaluation

The best performing model was the CNN though there was not a significant difference between the methods evaluated. While we limited the time spent on hyper-parameter tuning of the ensemble approach, it was interesting that it fared the worst in our evaluation. The AUC-ROC score enabled us to compare modeling approaches but does not reflect the performance in production. When de-

Number of Tags Shown	Accuracy
Top 5 labels	0.95
Top 3 labels	0.76
Top 1 labels	0.47

Table 3: TF-ILF+SVM Model Accuracy

ployed, the model shows users the top five label predictions. Users can pick any subset of those labels to apply them to the question being reviewed. To get a sense of accuracy in production, we log how many times we cover all relevant labels in the top N predictions as shown in Table 3.

6 Impact Analysis

We are currently logging editor activity and calculating metrics to perform a thorough impact analysis. Initial estimates show that time spent on labeling questions with NCLEX tags went from a few minutes pre-machine learning to less than one minute after our solution was deployed. There are tens of thousands of questions in Wolters Kluwer products like PrepU and CoursePoint and more content being generated every year. This impact is therefore significant, measuring several hours and potentially up to $100,000 or more savings annually.

Editors have responded very positively and regularly use machine learning label suggestions in their current workflow. That said, it will take some time for them to accept a completely automated process. Perhaps more importantly, subject matter experts have assessed that the consistency and quality of labels assigned to questions increase with the model suggestions. Nursing content editors often apply labels based on their personal understanding of content, which is sometimes subjective. There may also be biases in selecting "convenient" labels when having to choose from a lengthy list in a complicated workflow. The predictive model provides consistent label suggestions which in turn results in more consistent labels being assigned.

7 Future Work

The class imbalance of this task motivates the potential use of *active machine learning*. Some labels have only been assigned to a handful of questions. For these labels, we may work with editors to find more exemplar questions or create new ones. These new questions can then be merged with training data and the model retrained to ameliorate effects of class imbalance. In active learning, this process is typically repeated in an iterative process to target problem areas for a model. By selectively labeling new questions and down sampling over represented labels, we can fine tune data for retraining models to improve overall accuracy. Active machine learning has specifically been used for multi-label text classification problems (Yang et al., 2009).

Another area for further study is the evaluation of more recent, deep, transformer models. Because there is a great deal of semantic similarity between questions, these models may not fare better than more traditional vectorization and classification techniques. We intend to evaluate this hypothesis in future work.

There are many different tag sets and taxonomies that can be used to label nursing education content. Tagging both content and questions supports more advanced features such as dynamic remediation and adaptive learning. For instance, when a student answers a question incorrectly, learning software can automatically provide links to learning materials that are related to that topics addressed in that question. We are actively investigating how tagging and organizing content can support various use cases for adaptive learning.

References

Ashutosh Adhikari, Achyudh Ram, Raphael Tang, and Jimmy Lin. 2019. DocBERT: BERT for Document Classification. *arXiv e-prints*, page arXiv:1904.08398.

Tal Baumel, Jumana Nassour-Kassis, Michael Elhadad, and Noémie Elhadad. 2017. Multi-label classification of patient notes a case study on ICD code assignment. *CoRR*, abs/1709.09587.

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

Hrayr Harutyunyan, Hrant Khachatrian, David C. Kale, Greg Ver Steeg, and Aram Galstyan. 2019. Multitask learning and benchmarking with clinical time series data. *Scientific Data*, 6(1).

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *CoRR*, abs/1508.01991.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751,

Doha, Qatar. Association for Computational Linguistics.

Ladislav Lenc and Pavel Král. 2017. Word embeddings for multi-label document classification. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 431–437, Varna, Bulgaria. INCOMA Ltd.

Andrew Mccallum and Kamal Nigam. 2001. A comparison of event models for naive bayes text classification. *Work Learn Text Categ*, 752.

Tomas Mikolov, Wen tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *HLT-NAACL*, pages 746–751. The Association for Computational Linguistics.

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

Sampo Pyysalo, F Ginter, Hans Moen, T Salakoski, and Sophia Ananiadou. 2013. Distributional semantics resources for biomedical text processing. *Proceedings of Languages in Biology and Medicine*.

Jesse Read, Luca Martino, Pablo M. Olmos, and David Luengo. 2015. Scalable multi-output label prediction: From classifier chains to classifier trellises. *Pattern Recognition*, 48(6):2096 – 2109.

Newton SpolaôR, Everton Alvares Cherman, Maria Carolina Monard, and Huei Diana Lee. 2013. A comparison of multi-label feature selection methods using the problem transformation approach. *Electron. Notes Theor. Comput. Sci.*, 292:135–151.

Jinzhong Xu, Jie Liu, and Xiaoming Liu. 2009. Research on topic relevancy of sentences based on hownet semantic computation. In *9th International Conference on Hybrid Intelligent Systems (HIS 2009), August 12-14, 2009, Shenyang, China*, pages 195–198. IEEE Computer Society.

Bishan Yang, Jian-Tao Sun, Tengjiao Wang, and Zheng Chen. 2009. Effective multi-label active learning for text classification. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 917–926.