

GritNet: Student Performance Prediction with Deep Learning

Byung-Hak Kim, Ethan Vizitei, Varun Ganapathi
Udacity
2465 Latham Street
Mountain View, CA 94040
{hak, ethan, varun}@udacity.com

ABSTRACT

Student performance prediction - where a machine forecasts the future performance of students as they interact with on-line coursework - is a challenging problem. Reliable early-stage predictions of a student's future performance could be critical to facilitate timely educational interventions during a course. However, very few prior studies have explored this problem from a deep learning perspective. In this paper, we recast the student performance prediction problem as a sequential event prediction problem and propose a new deep learning based algorithm, termed GritNet, which builds upon the bidirectional long short term memory (BLSTM). Our results, from real Udacity students' graduation predictions, show that the GritNet not only consistently outperforms the standard logistic-regression based method, but that improvements are substantially pronounced in the first few weeks when accurate predictions are most challenging.

Keywords

Student performance prediction, Deep learning for education, Educational data mining, Learning analytics

1. INTRODUCTION

Education is no longer a one-time event but a lifelong experience. One reason is that working lives are now so lengthy and fast-changing that people need to keep learning throughout their careers [3]. While the classic model of education is not scaling to meet these changing needs, the wider market is innovating to enable workers to learn in new ways. Massive open online courses (MOOCs), offered by companies such as Udacity and Coursera, are now focusing much more directly on courses that make their students more employable. At Coursera and Udacity, students pay for short programs that bestow *microcredentials* and *Nanodegrees* in technology-focused subjects such as self-driving cars and Android. Moreover, universities are offering online degrees to make it easier for professionals to access opportunities to develop their skills (e.g., Georgia Tech's Computer Science Master's degree).

However, broadening access to cutting-edge vocational subjects does not naturally guarantee student success [2]¹. In a classic classroom, where student numbers are limited, various dimensions of interactions enable the teacher to quite effectively assess an individual student's level of engagement,

¹HarvardX and MITx have reported that only 5.5% of people who enroll in one of their online courses earn a certificate.

and anticipate their learning outcomes (e.g., successful completion of a course, course withdrawals, final grades). In the world of MOOCs, the significant increase in student numbers makes it impractical for even experienced *human* instructors to conduct such individual assessments. An automated system, which accurately predicts how students will perform in real-time, could possibly help in this case. It would be a valuable tool for making smart decisions about when to make live educational interventions during the course (and with whom), with the aim of increasing engagement, providing motivation and empowering students to succeed.

The student performance prediction problem has been partly studied within the learning analytics and educational data mining communities in the form of the student dropout (or completion) prediction problem (which is an important subclass problem of the student performance prediction problem). Most previous works can be divided into two approaches:

- The first traditional approach principally relies on generalized linear models, including logistic regression, linear SVMs and survival analysis (see [10] for a thorough summary). Each model considers different types of behavioral and predictive features extracted from various raw activity records (e.g., clickstream, grades, forum, grades).
- The second emerging approach involves an exploration of neural networks (NN). Few prior works explore deep neural network (DNN) model [10], recurrent neural network (RNN) model [6] and convolutional neural networks (CNN) followed by RNN [9]. However, all of these new models, so far, have shown primitive performance. This is mainly because the models still rely on feature engineering to reduce input dimensions which appears to limit one to develop larger (i.e., better) NN models.

Student activity records collected from different courses often have various lengths, formats and content, so that features that are effective in one course might not be so in another. Even carefully designed feature dimensions are usually constrained to be small². Both of these deficiencies produce inputs that are, so far, too restricted to tap the full

²In past works, DNN of width 5 [10] and LSTM of 20 cell dimensions [6] are used.

benefits of sequential deep learning models. To avoid the deficiencies of prior works, GritNet takes students’ learning activities across time as raw input (see Section 2.3.1) and (implicitly) searches for parts of an event embedding sequence that are most discriminative to predicting a student’s performance without having to engineer those parts as an (explicit) input feature (see Section 2.3.2).

In the remainder of this paper, we introduce the basic GritNet model in Section 2, followed by the Udacity data and training discussions in Section 3. In Section 4, we demonstrate the performance of GritNet via experimental results and give conclusions in Section 5.

2. GritNet

2.1 Problem Formulation

The task of predicting student performance can be expressed as a sequential event prediction problem [8]: given a past event sequence $\mathbf{o} \triangleq (o_1, \dots, o_T)$ taken by a student, estimate likelihood of future event sequence $\mathbf{y} \triangleq (y_{T+D}, \dots, y_{T'})$ where $D \in \mathbb{Z}_+$.

In the form of online classes, each event o_t represents a student’s action (or activities) associated with a time stamp. In other words, o_t is defined as a paired tuple of (a_t, d_t) . Each action a_t represents, for example, “a lecture video viewed”, “a quiz answered correctly/incorrectly”, or “a project submitted and passed/failed”, and d_t states the corresponding (logged) time stamp.

Then, log-likelihood of $p(\mathbf{y}|\mathbf{o})$ can be written as Equation 1, given fixed-dimensional embedding representation v of \mathbf{o} .

$$\log p(\mathbf{y}|\mathbf{o}) \simeq \sum_{i=T+D}^{T'} \log p(y_i|v) \quad (1)$$

The goal of each GritNet is, therefore, to compute an individual log-likelihood $\log p(y_i|v)$, and those estimated scores can be simply added up to estimate long-term student outcomes.

2.2 Baseline Model

In order to assess how much added value is brought by the GritNet, logistic regression is used as a baseline model. Here, we use the bag of words (BoW) model to represent each student’s past event sequence \mathbf{o} . After transforming all students’ activities into a BoW, we count the number of times each unique activity appears in \mathbf{o} .

Let fixed-dimensional feature representation v of \mathbf{o} be an N -dimensional feature vector where $v_j \in \mathbb{Z}_{\geq 0}$. Given v , logistic regression models $\log p(y_i|v)$ as follows:

$$\log p(y_i = 1|v; \theta) = \frac{1}{1 + \exp(-\theta^T v)}, \quad (2)$$

where $\theta \in \mathbb{R}^N$ are the parameters of the logistic regression model. For M training instances $\{(v^{(k)}, y^{(k)})\}_{k=1}^M$, L_2 regularized logistic regression finds the parameters θ that solve the following optimization problem:

$$\arg \max_{\theta} \sum_{k=1}^m \log p(y^{(k)}|v^{(k)}; \theta) + \alpha \|\theta\|_2. \quad (3)$$

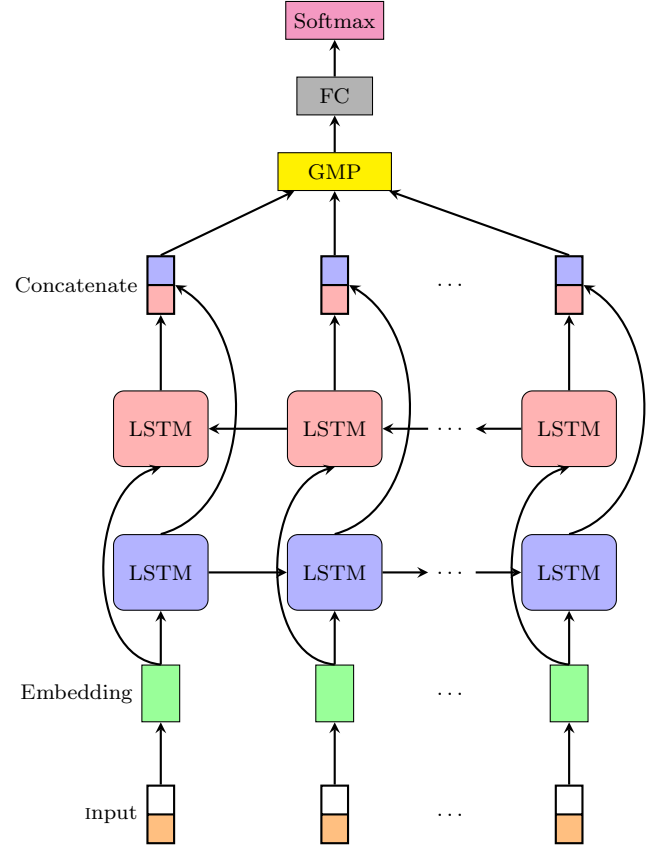


Figure 1: Architecture of a GritNet for the student performance prediction problem as described in Section 2.3.

Often, it will be convenient to consider L_1 regularized logistic regression instead of Equation 2 to handle irrelevant features [7]. We noticed even simpler feature selection methods (e.g., Chi-Square score based), combined with L_2 regularized logistic regression, provides similar results as the L_1 based.

2.3 GritNet Architecture

2.3.1 Input Representation

In order to feed students’ raw event records into the GritNet, it is necessary to encode the time-stamped logs (ordered sequentially) into a sequence of fixed-length input vectors³. We do this simply by *one-hot encoding*. A one-hot vector $\mathbb{1}(a_t) \in \{0, 1\}^L$, where L is the number of unique actions and j -th element defined as:

$$\mathbb{1}(a_t)_j \triangleq \begin{cases} 1 & \text{if } j = a_t \\ 0 & \text{otherwise} \end{cases}, \quad (4)$$

is used to distinguish each activity a_t from every other. Then, we connect one-hot vectors of the same student into a long vector sequence to represent the student’s whole sequential activities in \mathbf{o} .

³GritNet does not need manual feature selections [6] or time-series input aggregations per normalized time intervals [9].

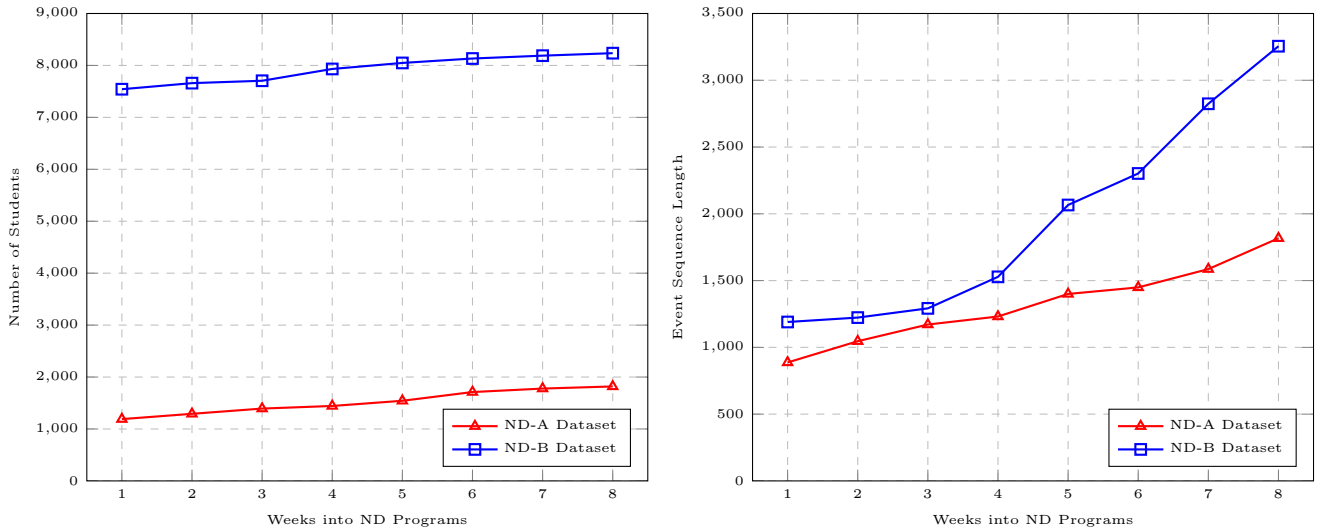


Figure 2: Student and event characteristics of two Udacity Nanodegree program datasets (ND-A and ND-B) by week: size of dataset (left) and maximum event sequence length (right). The reason the number of students climbs over time is that we only include students who have interacted with their mentor, so if they do not interact in the first couple weeks they are not included early on (left). As a student progresses, the accumulated event sequence gets longer (right).

While this encoding method preserves ordering information, in contrast to the BoW method (see Section 2.2), it has a limitation in capturing students’ learning speed. Varying learning speed is an important piece of time-dependent information, reflecting a student’s progress and/or the course content’s difficulty. Since directly employing each time stamp d_t will increase the input space too fast, we define the discretized time difference between adjacent events as⁴:

$$\Delta_t \triangleq d_t - d_{t-1}. \quad (5)$$

Then, one-hot encode Δ_t into $\mathbb{1}(\Delta_t)$ and connect them with the corresponding $\mathbb{1}(a_t)$ to represent $\mathbb{1}(o_t)$ as:

$$\mathbb{1}(o_t) \triangleq [\mathbb{1}(a_t); \mathbb{1}(\Delta_t)]. \quad (6)$$

Lastly, we pre-pad the output sequences shorter than the maximum event sequence length (of a given training set) with all $\mathbf{0}$ vectors.

2.3.2 Model Architecture

The core of our GritNet model is the embedding [1], BLSTM [4] and GMP [5] layers trained to ingest past student events and predict a log likelihood of a future one. The first embedding layer⁵ learns an embedding matrix $\mathbf{E}^o \in \mathbb{R}^{E \times |O|}$, where E and $|O|$ are the embedding dimension and the number of unique events (i.e., input vector $\mathbb{1}(o_t)$ size), to convert an input vector $\mathbb{1}(o_t)$ onto a low-dimensional embedding v_t de-

finied as:

$$v_t \triangleq \mathbf{E}^o \mathbb{1}(o_t). \quad (7)$$

This event embedding v_t is then passed into the BLSTM and the output vectors are formed by concatenating each forward and backward direction outputs. Next, a GMP layer is added before the output layer. With the GMP layers, GritNet learns to focus the most relevant part of the event embedding sequence while ignoring the rest. This GMP operation seems crucial in boosting prediction power, particularly for imbalanced data provided without any feature engineering⁶.

The GMP layer output is, ultimately, fed into a fully-connected layer and a softmax (i.e., sigmoid) layer sequentially to calculate the log-likelihood $\log p(y_i|v)$. The complete GritNet architecture is illustrated in Figure 1.

3. DATA AND TRAINING

3.1 Udacity Data

We benchmarked our methods on the student datasets of two Udacity Nanodegree (ND) programs: ND-A and ND-B. These two ND programs were selected specifically because they diverge from each other along many axes. For example, ND-A curriculum has a lower expectation of prior technical knowledge and a relatively higher graduation rate than ND-B.

⁴For the Udacity data described in Section 3.1, we use day to represent inter-event time intervals.

⁵With an embedding layer which provides a dense representation for an event, GritNet achieves an improved performance on the Udacity dataset. Furthermore, after training, similar events appear to be closer in the embedding event space.

⁶We empirically find that vanilla BLSTM (without the GMP layer) on the (imbalanced) Udacity datasets does not yield comparable results as shown in Figure 3. A GMP layer appears to combat this imbalanced data issue effectively by ensuring training errors back-propagate only to the network weights corresponding to the most discriminative part within the event embedding sequence.

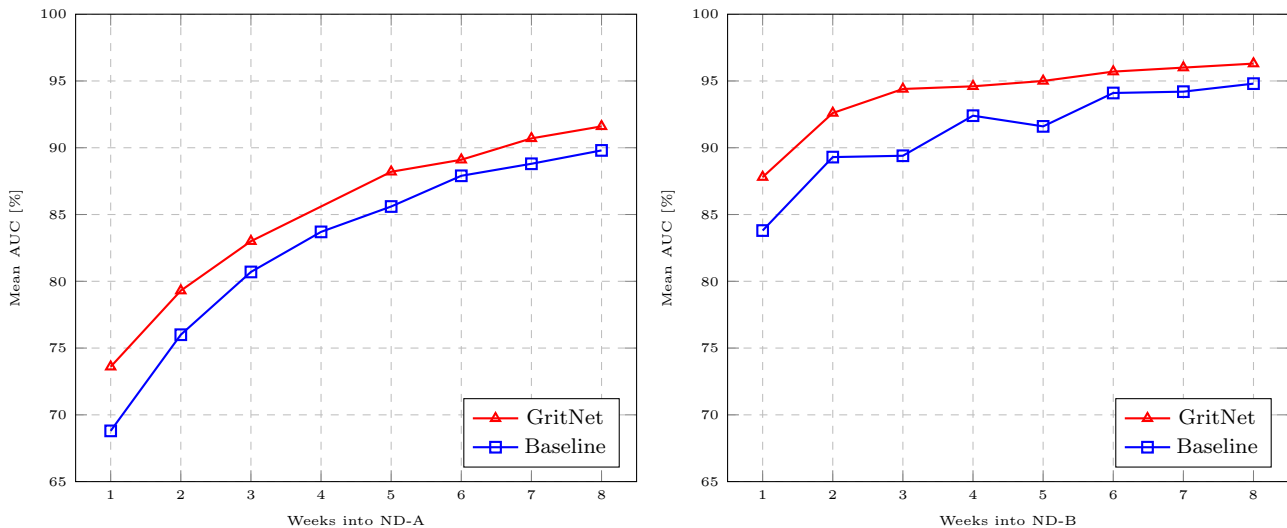


Figure 3: Student graduation prediction accuracy comparisons of the GritNet vs baseline models in terms of mean AUC (over all five folds) on two Udacity Nanodegree program datasets: ND-A (left) and ND-B (right). GritNet provides 5.3% abs (7.7% rel) accuracy improvements at week 1 for ND-A dataset (left). Notice that for ND-B dataset, the baseline model requires eight weeks of student data to achieve the same performance as the GritNet is able to achieve with only three weeks of student data (right).

In both programs, graduation is defined as completing each of the required projects in a ND program curriculum with a passing grade. When a user officially graduates, their enrollment record is annotated with a time stamp, so it was possible to use the presence of this time stamp as the target label. Users have to graduate before 2017-09-30 to be considered as successfully graduated.

Each ND program’s curriculum contains a mixture of video content, written content, quizzes, and projects. Note that it is not required to interact with every piece of content or complete every quiz to graduate. See below for detailed characteristics of each dataset used for this study.

- ND-A Dataset:** From the students who enrolled in ND-A program (from 2017-03-07 to 2017-09-30), we selected 1,853 students who had actively engaged with their classroom mentor (believing these to be the students exhibiting full engagement with the curriculum overall). This set of 1,853 students includes 777 students who graduated, yielding a graduation rate of 41.9%. The length of each student’s events streams ranges from 0 to 4,175 events, with an average of 536 events. The curriculum for ND-A program contains 9 projects, 1,025 unique content pages to visit, and 77 quizzes to attempt.
- ND-B Dataset:** As prescribed above, we selected 8,301 students who actively engaged with their classroom mentor from the students who enrolled in ND-B program (from 2016-06-20 to 2017-09-30). This set of 8,301 students includes 1,005 students who graduated, yielding a graduation rate of 12.1%. The length of each student’s event streams ranges from 1 event to 4,554 events, with an average of 242 events. The curriculum for ND-B program is composed of 6 projects, 668

unique content pages, and 347 quizzes.

For both datasets, an event represents a user taking a specific action (e.g., watching a video, reading a text page, attempting a quiz, or receiving a grade on a project) at a certain time stamp. Some irrelevant data is filtered out during preprocessing, for example, events that occur *before* a user’s official enrollment as a result of a free-trial period. It should be noted that no personally identifiable information is included in this data and student equality is determined via opaque unique ids.

3.2 Training

We learned that the GritNet models are fairly easy to train. The training objective is the negative log likelihood of the observed event sequence of student activities under the model. The binary cross entropy loss is minimized⁷ using stochastic gradient descent on mini-batches.

In our experiment, the BLSTM with forward and backwards LSTM layers containing 128 cell dimensions per direction is used. Embedding layer dimension was grid-searched for the best parameters based on the dataset: from 1024 to 3584 for ND-A set and from 1024 to 5120 for ND-B set. A dropout rate, ranged from 10 to 20%, applied to the BLSTM output and worked well for both datasets to prevent overfitting during training with a mini-batch size of 32.

For both baseline and GritNet models, we trained a different model for different weeks, based on students’ week-by-week event records, to predict whether each student was likely to

⁷In this case, minimizing the binary cross entropy is equivalent to maximizing the log likelihood.

graduate. Figure 2 shows the number of students and the (longest) event sequence length of a student, both observed at each week.

4. PREDICTION PERFORMANCE

4.1 Evaluation Measure

To demonstrate the benefits of the GritNet, we focused on student graduation prediction. Since the true binary target label (1: graduate, 0: not graduate) is imbalanced (i.e., number of 0s outweighs number of 1s), accuracy is not an appropriate metric. Instead, we used the Receiver Operating Characteristic (ROC) for evaluating the quality of the GritNet’s predictions. An ROC curve was created by plotting the true positive rate (TPR) against the false positive rate (FPR). In this task, the TPR is the percentage of students who graduate, which the GritNet labels positive, and the FPR is the percent of students who do not graduate, which the GritNet incorrectly labels positive.

The accuracy of each system’s prediction was measured by the area under the ROC curve (AUC) which scores between 0 and 100% (the higher, the better) – with random guess yielding 50% all the time. We used 5-fold student level cross-validation, while ensuring each fold contained roughly the same proportions of the two groups (graduate and non graduate) of students.

4.2 Results

For fair comparisons, the baseline performance was optimized by sweeping α values (in Equation 3) at each week⁸. The GritNet also required slight hyper-parameter optimization (e.g., embedding dimension as prescribed in Section 3.2) for the optimal accuracy at each week.

We have shown that the GritNet really does improve the student graduation prediction accuracy across weeks. From the prediction results on both Udacity datasets in Figure 3, we clearly see that the performance is similar between the baseline and GritNet models after receiving eight weeks of data about a given student. However, the GritNet is able to achieve significant prediction-quality improvements within the first few weeks of the student experience.

Specifically, the GritNet was able to attain superior performance by more than 5.0% abs on both ND-A dataset (at week 1) and ND-B dataset (at week 3). Moreover, on ND-B dataset, the baseline model required a wait of two months to reach the prediction accuracy that the GritNet showed within three weeks. We believe this is a crucial advantage of the GritNet, creating a quickly adaptable but accurate metric to estimate long-term student outcomes to accelerate the student feedback loop (which typically takes a few months from enrollment to iterate).

5. CONCLUSION

In this paper, we have successfully applied deep learning to the challenging student performance prediction problem which, so far, has not been fully exploited. In contrast to prior work, we formulated the problem as a sequential event

prediction problem, introduced a new algorithm called the GritNet to tackle the problem, and demonstrated the superiority of the GritNet using student data from Udacity’s Nanodegree programs.

Two novel properties of the GritNet are that (1) it does not need any feature engineering (it can learn from raw input) and (2) it can operate on any student event data associated with a time stamp (even when highly imbalanced). For future work, we anticipate that incorporating indirect data (e.g., student board activity, interactions with mentors) into the GritNet will potentially further improve the GritNet’s impressive performance.

6. REFERENCES

- [1] Y. Bengio, R. Ducharme, and P. Vincent. A neural probabilistic language model. In *Advances in Neural Information Processing Systems 13 (NIPS 2000)*, pages 932–938, 2001.
- [2] I. Chuang and A. Ho. HarvardX and MITx: Four years of open online courses, Fall 2012-Summer 2016. *SSRN Electronic Journal*, 2016.
- [3] Economist. Equipping people to stay ahead of technological change - Lifelong learning. *A Special Report On Lifelong Learning: How to Survive in the Age of Automation*, 2017.
- [4] A. Graves and J. Schmidhuber. Framewise phoneme classification with bidirectional LSTM networks. In *2005 International Joint Conference on Neural Networks (ICJNN’05)*, pages 23–43, 2005.
- [5] J. D. Keeler, D. E. Rumelhart, and W. K. Leow. Integrated segmentation and recognition of hand-printed numerals. In *Advances in Neural Information Processing Systems 3 (NIPS 1990)*, pages 557–563, 1991.
- [6] F. Mi and D.-Y. Yeung. Temporal models for predicting student dropout in massive open online courses. In *Proceedings of 15th IEEE International Conference on Data Mining Workshop (ICDMW 2015)*, pages 256–263, Atlantic City, New Jersey, 2015.
- [7] A. Y. Ng. Feature selection, L1 vs. L2 regularization, and rotational invariance. In *Proceedings of the 21st International Conference on Machine Learning (ICML 2004)*, pages 78–85, Banff, Alberta, Canada, 2004.
- [8] C. Rudin, B. Letham, A. Salieb-Aouissi, E. Kogan, and D. Madigan. Sequential event prediction with association rules. In *24th Annual Conference on Learning Theory (COLT 2011)*, pages 615–634, 2011.
- [9] W. Wang, H. Yu, and C. Miao. Deep model for dropout prediction in moocs. In *Proceedings of the 2nd International Conference on Crowd Science and Engineering (ICCSE 2017)*, pages 26–32, Beijing, China, 2017.
- [10] J. Whitehill, K. Mohan, D. Seaton, Y. Rosen, and D. Tingley. Delving deeper into mooc student dropout prediction. *arXiv preprint arXiv:1702.06404*, 2017.

⁸The optimized results are quite strong such that initially explored NN models (e.g., DNN, CNN-BLSTM) on the same BoW input features did not yield big win over the baseline.