# UNSUPERVISED RECOMMENDATION AND DISCOVERY OF AN EDUCATION MARKETPLACE

**Akshay Budhkar,**
Georgian Partners, 2 St Clair Ave W #1400, Toronto, ON M4V 1L5,
abudhkar@georgianpartners.com

**Will Bradbeer,**
Top Hat, 151 Bloor St W Suite 200, Toronto, ON M5S 1S4,
will.bradbeer@tophatmonocle.com

**Parinaz Sobhani,**
Georgian Partners, 2 St Clair Ave W #1400, Toronto, ON M4V 1L5,
psobhani@georgianpartners.com

## ABSTRACT

Active learning is an effective teaching tool that can significantly benefit from large amounts of academic content. This content exists but is underutilized by most professors due to the friction of retrieving it. In this work, we design a recommendation system to enhance the discoverability of content on an active learning platform. The system explores several representation learning models and uses proxy labels to optimize hyperparameters. Finally, the system is tested with subject matter experts to ensure the validity of the recommendations before deploying it to production.

## 1 INTRODUCTION

*Active Learning*, in education, can be defined as any activity that aids students by engaging them in the learning process. It has shown to be a useful tool for improving academic success across a variety of methods, disciplines and education levels in (Prince, 2004). However, implementation is difficult as it requires flexible technology and large amounts of academic content. In particular, many questions are needed to provide the recommended brief, but frequent, activities of a lecture.

Top Hat is a turn-key software providing thousands of professors the ability to implement active learning in their classroom. This software connects to a marketplace (MP) with paid content and Open Educational Resources (OER). Use of the MP is essential to the success of active learning as it is tedious to create the number of questions required manually. Currently, this MP is underutilized due to the challenges of finding the relevant content. This creates an opportunity to improve the implementation of active learning in institutions using Top Hat's software by recommending useful questions from the MP, leading to improved student performance.

This project focuses on making recommendations of Psychology questions on the MP, with the goal of improving the discoverability of this content. It is the basis for future recommendation systems that would engage all content types and extend to all subjects on the platform. This system could be extended to Mastery Learning, as described in (Kulik et al., 1990), by recommending questions similar to those that students perform poorly on.

### 1.1 DATASET EXPLORATION

Figure 1 shows a distribution of the number of questions generated weekly by the professors across psychology in the last two years. The number of questions that professors created per week has a mean of 130, a standard deviation of 530, and a median of 11 questions.

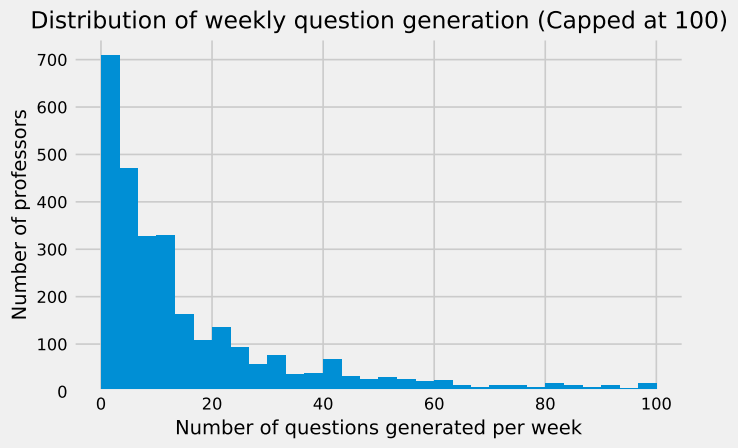


Figure 1: Distribution of Number of Weekly Question generated across courses in Psychology.

**Context** is defined as the section of text used as an input to the unsupervised model, for both training and inference. The combined text from the questions already created by the professor forms the context. This setup assumes that the professors want new questions similar to those they have recently created.

Blocks of up to 30 professor-generated same-week questions were used as the context to train the representation models. This structure covers more than 70% of the corpus and excludes any weeks with more than 30 questions, as these were deemed to be undesirable outliers (for example, it excludes cases where there are mass imports from the MP for demo purposes). During inference, any weeks with more than 30 questions were broken down into multiple groups of 30, with each group getting served its set of recommended MP questions. For multiple choice questions, the context also included the text of the answer.

### 1.2 Proxy Label

For purposes of hyper-parameter tuning, model selection, and performance evaluation, this study used historical MP import data. This data is from professors who have manually imported questions from the MP. Courses were segmented into time windows of determined size and the questions created by these professors during that period were used as the context. Actual imported questions were then used to evaluate the model recommendations.

The following metrics were used to measure the performance of the recommendation system (here $k$ is the number of top questions that are served as a recommendation):

**Precision at k**: (# of recommended items @k that are relevant) / (# of recommended items @k).

**Recall at k**: (# of recommended items @k that are relevant) / (total # of relevant items).

**Modified Precision at k**: (# of recommended items @k that are relevant) / min((# of recommended items @k), (# of items that are relevant)).

The Modified Precision at k metric handled the typical scenarios where there are very few relevant items and was therefore chosen as the metric to optimize during parameter tuning.

All reported metrics set $k$ to 10.

The window size was empirically chosen to be two months, by studying a single model's performance on four different window sizes as seen in Table 1 below. This setup leads to **320** labeled instances across all Psychology courses.

Table 1: Performance of a tf-idf model on two years of psychology import data using different window sizes. The window size is chosen to optimize modified precision as all the time horizons have a comparable number of labels. Random modified precision is measured by sampling $k$ items, averaged over **100** runs.

| Time Horizon | Number of Labels | Precision | Modified Precision | Recall | Random Modified Precision |
|---|---|---|---|---|---|
| 2 weeks | 382 | 8.80% | 12.17% | 7.04% | 1.26% |
| 1 Month | 346 | 10.43% | 14.58% | 7.87% | 2.04% |
| 2 Months | 320 | 10.27% | **14.76%** | 8.60% | 1.59% |
| 4 Months | 302 | 7.60% | 10.77% | 5.80% | 1.98% |

## 2 Representation Models

For the comparison of representation models, a random selection model acted as a baseline. For every label, $k$ items are randomly sampled and the metrics are averaged over **1000** runs.

A term frequency-inverse document frequency model or **tf-idf** is a numerical statistical model that learns how important a word is in documents across the corpus. This value increases proportionally with the frequency of a word per document and decreases proportionally with the frequency across the corpus.

(Mikolov et al., 2013)'s **word2vec** learns distributed representations for words by training a shallow three-dimensional neural network to predict context words given the current word (or vice versa). Document representations are obtained by averaging the word2vec representations for each word in that document, ending up with the same number of dimensions as the word vectors.

Thirdly, (Le & Mikolov, 2014)'s **doc2vec** approach learns distributed representations for documents directly, by appending IDs to the documents and learning context word predictions (similar to word2vec) for the specific document at hand.

The tf-idf vector representation is trained using the sklearn[1] library and the latter two models are trained using the optimized gensim[2] library.

This study also explored Facebook's fastText embedding (Bojanowski et al., 2017) to include subword information, and Google's Universal Sentence Encoder (Cer et al., 2018) to leverage transfer learning from everyday NLP tasks, but both these models, with unoptimized hyperparameters, performed significantly worse than those mentioned above. The average fastText embedding model (default gensim setting and with *dim*=500, *window_size*=10, *iter*=10) had precision, modified precision and recall of 1.53%, 1.53% and 0.2% respectively. The default Google Universal Sentence Encoder had precision, modified precision and recall of 4.48%, 5.39% and 0.28% respectively. We exclude these models from the next steps.

### 2.1 Hyperparameter Tuning

(Bergstra et al., 2013)'s **hyperopt** learns a meta Bayesian model to optimize hyperparameters while optimizing for a specified loss function. Hyperopt's python library[3] was used to tune pre-determined parameter spaces. Each model was run through 500 trials to find the best-performing parameters, maximizing the modified precision metric discussed in Section 1.2.

## 3 Recommendation Results

### 3.1 Model Performance

The best performing models on the proxy label (Section 1.2) are reported in Table 2. All three models outperform the random baseline. Doc2vec and tf-idf perform better than average word2vec.

[1] https://scikit-learn.org/stable/index.html
[2] https://radimrehurek.com/gensim/
[3] https://github.com/hyperopt/hyperopt

Table 2: Model Performance on the Proxy Label. Only the best performing hyper-parameter tuned model scores are reported here.

| Model | Precision | Modified Precision | Recall |
|---|---|---|---|
| Random | 1.62% | 1.69% | 0.3% |
| TF-IDF | 8.34% | **10.36**% | 4.5% |
| Average word2vec | 6.84% | 8.62% | 2.45% |
| Doc2vec | 6.68% | **9.22**% | 4.6% |

### 3.2 Internal A/B Testing

**50** examples were chosen at random and compared the performance of the best tf-idf model to suggesting random questions from the MP (28 *model*, 22 *random* - sampled from a 50/50 distribution), to ensure that the proxy results translate to relevant recommendations. Each context was recommended ten questions.

Several experts from the Top Hat team scored each recommended question (50 examples * 10 recommendations each = **500** recommendations) - giving them a score of 0 if not relevant, 1 if tangentially relevant and 2 if relevant.

The scorers distributed the work among themselves, and one to three people scored each example. Six examples (three each from model and random) were flagged as having not enough or irrelevant context by at least one examiner, and hence, excluded in the final results.

Figure 2 shows the distribution of the average scores for both the random and model cases. There are 190 total random examples, and 250 total model examples.

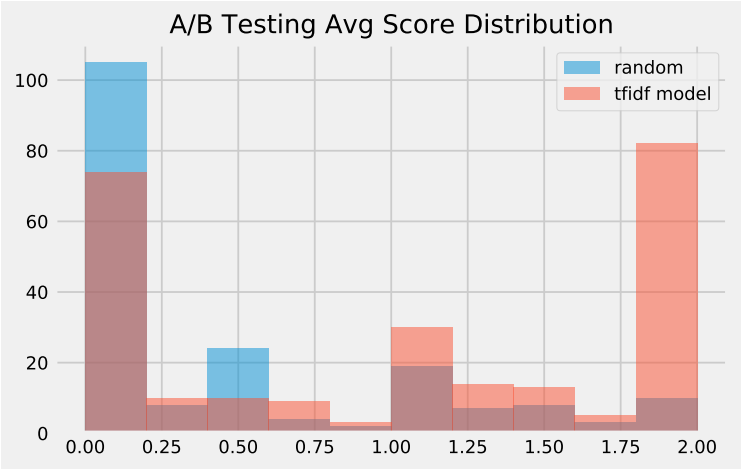


Figure 2: Average score distribution of the Internal A/B Test.

Table 3 reports the statistics of the scores across the random and model examples. The average model score is 2.33 times better than random. More than half of the average scores for the random model are 0 and more than half of the maximum scores of the ML model are 2, implying that at least one of the experts found the recommendation to be relevant for the majority of the contexts. The experts also described the differences in the two models as *obvious* while testing.

Future work involves deploying the tf-idf and the doc2vec models in production and monitoring their performance by comparing the professor interactions to the baselines of random selection, most popular questions, and status quo (no recommendations).

Table 3: Internal A/B test statistics. Average, min and max taken over the multiple scorers per recommended example.

| Score Type | Random | Model |
|---|---|---|
| Mean of Avg. | 0.44 | **1.03** |
| Median of Avg. | 0.0 | **1.0** |
| Mean of Min | 0.23 | **0.796** |
| Mean of Max | 0.6421 | **1.24** |
| Median of Max | 0.0 | **2.0** |

## 4 CONCLUSION

In this work, we proposed a recommendation system for an active learning platform to improve content discoverability. We trained multiple unsupervised representation models on an existing corpus of data. We leveraged historical import data to choose the best models and do bayesian hyperparameter tuning. Finally, a team of experts rated the performance on 500 recommendations to verify the validity of the proxy label. This work sets an example for building recommender systems with minimal or no training data, by leveraging proxy labels and internal testing to ensure model validity before deploying to production.

## REFERENCES

James Bergstra, Daniel Yamins, and David Daniel Cox. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. 2013.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*, 2018.

Chen-Lin Kulik, James A. Kulik, and Robert L. Bangert-Drowns. Effectiveness of mastery learning programs: A meta-analysis. *Review of Educational Research*, 60(2):265–299, 1990.

Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pp. 1188–1196, 2014.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

Michael Prince. Does active learning work? a review of the research. *Journal of Engineering Education*, (July):223–231, 2004.