Human-level Multiple Choice Question Guessing Without Domain Knowledge: Machine-Learning of Framing Effects

Patrick Watson
IBM Research
Yorktown Heights, New York
pwatson@us.ibm.com

TengFei Ma
IBM Research
Yorktown Heights, New York
tengfei.ma1@ibm.com

Ravi Tejwani
IBM Research
Yorktown Heights, New York
rtejwan@us.ibm.com

Maria Chang
IBM Research
Yorktown Heights, New York
maria.chang@us.ibm.com

JaeWook Ahn
IBM Research
Yorktown Heights, New York
jaewook.ahn@us.ibm.com

Sharad Sundararajan
IBM Research
Yorktown Heights, New York
sharads@us.ibm.com

ABSTRACT

The availability of open educational resources (OER) has enabled educators and researchers to access a variety of learning assessments online. OER communities are particularly useful for gathering multiple choice questions (MCQs), which are easy to grade, but difficult to design well. To account for this, OERs often rely on crowd-sourced data to validate the quality of MCQs. However, because crowds contain many non-experts, and are susceptible to question framing effects, they may produce ratings driven by guessing on the basis of surface-level linguistic features, rather than deep topic knowledge. Consumers of OER multiple choice questions (and authors of original multiple choice questions) would benefit from a tool that automatically provided feedback on assessment quality, and assessed the degree to which OER MCQs are susceptible to framing effects. This paper describes a model that is trained to use domain-naive strategies to guess which multiple choice answer is correct. The extent to which this model can predict the correct answer to an MCQ is an indicator that the MCQ is a poor measure of domain-specific knowledge. We describe an integration of this model with a front-end visualizer and MCQ authoring tool.

CCS CONCEPTS

• **Applied computing → Learning management systems**;

KEYWORDS

OER, MCQs, Deep Learning, Blind Guessing

ACM Reference Format:
Patrick Watson, TengFei Ma, Ravi Tejwani, Maria Chang, JaeWook Ahn, and Sharad Sundararajan. 2018. Human-level Multiple Choice Question Guessing Without Domain Knowledge: Machine-Learning of Framing Effects. In *Proceedings of The 2018 Web Conference Companion (WWW'18 Companion).* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3184558.3186340

WWW'18 Companion, April 23–27, 2018, Lyon, France

ACM ISBN 978-1-4503-5640-4/18/04..
https://doi.org/10.1145/3184558.3186340

1 INTRODUCTION

Multiple choice questions (MCQs) are ubiquitously used for assessing learners and are easy to grade, but hard to design well. Open educational resources (OER) have provided educators and researchers with a wide variety of freely accessible, openly licensed, MCQs, which reduces the burden of having to design them. Many on line communities exist for exchanging OER assessments.[1][2][3] However, the quality of the MCQs shared in these communities is not guaranteed. Potential users of these materials are often left having to adapt them or re-write them entirely. This dilutes the value of OER resources and introduces variations among exams that can harm validity [5, 9]. These risks are familiar to the expert psychometricians who take pains to ensure their tests are well validated, but as OER resources on the web allow more and more educators to design their own tests, there is a critical need to provide tools to ensure that tests are fair, valid, and accurate assessments of knowledge.

To this end, we present a web-based tool that visualizes the output from a machine-learning based model trained to guess strategically on a corpus multiple choice questions based solely on domain-general linguistic features. We specifically exclude any topic-knowledge beyond vocabulary knowledge [7, 11] . Our goal is to help identify "gameable" questions within a well-validated and high-quality bank of OER MCQs. Because our model lacks domain-specific knowledge, it allows us to filter OER MCQs to identify how much they rely upon domain-specific knowledge, and how much performance on them could be driven by surface-level linguistic features.

2 NAIVE-FEATURE BASED GUESSING

Learners who lack necessary domain knowledge to complete an assessment fail gracefully, falling back on plausible guessing strategies such as choosing the longest answer, making logical comparisons among answers, and using surface-level linguistic similarity between sentences. We modeled this "naive features" guessing behavior by training a classifier to identify the correct answer solely on the basis of textual and linguistic features derived from domain-general language corpora. Additionally, we trained a classifier to

[1] www.oercommons.org
[2] https://sparcopen.org/
[3] https://open4us.org/find-oer

estimate the distribution of answers across alternatives to model the guessing behavior of a population of students.

We constructed a classifier f that is trained to select from the among the alternative responses of a multiple-choice item $A_i \ni [1, 2, ..n]$, based on the linguistic features of stem $L(s)$, the linguistic features of each alternative $L(A_i)$, and of the pair-wise relations between each pair of alternatives and between the stem and each alternative.

(1) Answer-Answer similarity:
   - For each answer, we calculated its similarity with all other answers. We use the average word embedding of all words in each answer as the vector representation of that answer, and then compute cosine similarity between these answer vectors. We create the vector representations using the pre-trained wikipedia word vectors from Glove[11]. Beyond word embedding based similarity, we also used Wordnet based similarity.
   - Mean and Variance. In addition to the raw similarity score, the static information of an answer compared to others could create a basis for selecting that answer alternative (i.e., an "odd answer out" could draw attention). Thus, we also used mean and variance of the answer similarities as features for each answer.

(2) Question-Answer similarity:
   - Relevance, i.e. the semantic similarity between a question and the answer.
   - Distance. Sometimes the average word embedding of a sentence fails to capture the importance of certain keywords. To account for keyword position we use a simplified version of the sentence distance calculation. Given the word vectors for the question $\{q_1, ..., q_n\}$ and the answer $\{a_1, ..., a_m\}$. The distance is calculated as follows: $dis = 1/n \sum_{i=1}^{n} \min_{j=1}^{m}(q_i, a_j)$

(3) Logical language
   - We additionally used presence of the logical keywords: "and","or","not," which are frequently used by question authors and serve to rule out certain answer combinations.

(4) Question and answer alternative length: the number of characters present in the question and answer alternatives, as well as the mean and variance between these. This allows us to identify which questions and answers contain the most details, which sometimes points to the correct answer.

To control for item-level variation among features, we scale each of these features by the variance at the level of the item as a whole (i.e., the variance of the stem and each alternative).

2.1 The Classifier

We then trained two variants of this classifier, using multi-layer perceptron (MLP). The first classifier (the "guesser") was trained to classify correct answers according to the ground-truth labeled answers. The second classifier (the "student distribution"), was trained to match the distribution of student answers. Both the classifiers are implemented in Keras [2]. The hyperparameters are as follows: 1. Above the embedding layer, we have two layers for MLP. The dimension for the first layer is 32, and the second layer is associated with a softmax function for prediction. 2. We used dropout for regularization with dropout rate 0.3. 3. The classifiers are trained by Adam with initial learning rate 0.01.

Table 1: Guessing Model Accuracy

Cross Validation	Mean Accuracy	Range
StratifiedKfold	52.88%	(+/- 5.57%)
Domain excluded	47.60%	(+/- 18.15%)
Balanced correct answers	43.93%	(+/- 5.63%)
AI2 questions	27.02%	(+/- 3.19%)

We trained both classifiers using two forms of cross validation, 10-fold random stratified cross validation using the scikit-learn package [10], and leave-one-group out cross validation, using each question topic (e.g., "cells", "plate tectonics") as the hold-back test set to ensure that the model was domain-general. This produced a 2x2 matrix of model variants (rows: guesser vs. student distribution, columns: random-stratified vs. domain-hold-back), performance ranges are reported in the Evaluation section.

2.2 The Data Set

Our data set consists of openly available MCQs from AAAS Project 2061.[4] Out of the 775 available questions, 26 contained pictorial or numerical answers and were excluded from this analysis. The remaining 749 questions covered topics in life sciences, physical science, earth science, and the nature of science. These questions were designed for use with middle school and high school students (typically ages 11-18 in the US). Each MCQ has a question stem, 4 answer options, student performance data, and misconception information (when available). In addition to provided the correct answer for each MCQ, this dataset also identifies the answer options that are most commonly selected by students. These questions were identified by AAAS as being especially conceptually complex and appropriate for assessing science knowledge.

However, this OER question set has an unbalanced distribution of correct responses across the four answer choices with an over-representation of "A" correct answers, and an under-representation of "D" correct answers. Students answers however, are uniformly distributed across the four responses. Since an unbalanced distribution enhances model guessing, we also trained a version of the model on a random sub-set of the questions with a uniform distribution of correct answers. As an additional control we trained and tested our model on the AI2 data set [1], a set of middle school science questions used to assess the domain knowledge of machine learning solutions. AI2's answer set is uniformly distributed.

3 EVALUATION

To evaluate our model of guessing, we examined the accuracy for guessing correct answers using linguistic features alone on the hold-back test set in our cross validation. We additionally examined both correct the modal answers of a hold-back set of multiple choice questions that were unseen in model training.

Importantly, the two different data sets (AAAS and AI2), produced radically different performance on naive-feature guessing.

[4] https://www.aaas.org/page/assessment-resources

Table 2: Feature-jackknife analysis

Feature Jackknife	accuracy	Range
All features included	52.88%	(+/- 5.57%)
Ans-Ans excluded	51.83%	(+/- 3.30%)
Ans-Quest excluded	50.80%	(+/- 4.53%)
Text features excluded	41.19%	(+/- 2.11%)
Ans-Quest and Text excluded	41.31%	(+/- 1.26%)

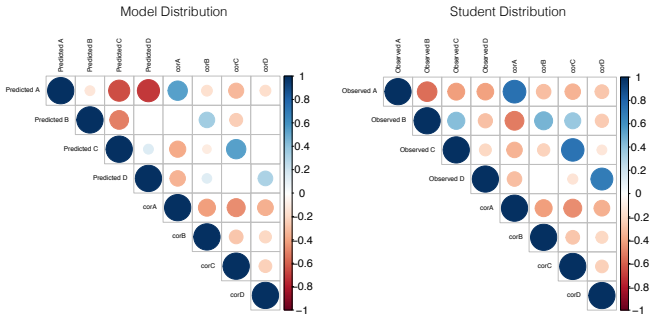

Figure 1: The model's predicted answer pattern, vs. the students observed answer pattern by correlation with correct answer.

In the OER AAAS set, the unbalanced distribution of answers and the presence of naive features resulted in a guessing rate above observed student performance (see Evaluation). While in the AI2 question set, our model performed only slightly above chance guessing. This highlights the importance of different selection criteria for inclusion in OER datasets. The AI2 question set was a challenge set for ML solutions and relies heavily on short, factual answers [1]. In contrast the AAAS data set was selected on the basis of conceptual difficulty [8], and is meant to assess student's understanding of the topic. This difference could help explain the large difference between guessing accuracies within these two question domains.

3.1 Feature-level contributions

We also evaluated the relative feature contributions via a jackknife analysis. We compared the relative contributions to accuracy of three categories of features 1) linguistic similarity among answer alternatives (ans-ans), 2) linguistic similarity between the question and the answer alternatives (ans-quest), and 3) low-level textual features (text features).

Both linguistic similarity and textual features were important contributors to model accuracy, although inclusion of both the Ans-Ans and the Ans-Quest features resulted in a small marginal improvement, suggesting that these two groups of features contained similar information.

3.2 Comparison to student performance

These numbers compare favorably to the sample of 6th-12th graders assessed using these questions. The mean accuracy (computed from the percent of the student population selecting the correct answer) was 44.2%. This was lower than our naive-features model's rate of correct guessing: 52.9%.

The variant of the model trained to match the student distribution also captured a broadly similar pattern of performance Figure 1. However, students exhibited some spatial clumping of their answers (tending to choose B when C was the correct answer and vice versa). This represents a possible domain of improvement of the model's predictive performance.

While the model's guessing based on naive features is below the state-of-the-art approach to middle school science questions using machine learning [1], it actually exceeds the observed performance of middle-schoolers. This has important consequences for the interpretation of the difficulty of items in this OER question set.

3.3 Chance-level Guessing with language knowledge

Item response theory estimates the probability of a correct answer via:

$$p(\Theta) = c + \frac{1-c}{1-e^{-a(\Theta-b)}} \tag{1}$$

Where Θ is an estimate of student skill, a is the item discrimination (i.e., the slope of the sigmoid), b is the item difficulty, and c is the base rate of correct answers assuming a uniform multinomial guessing distribution. We extend this by identifying two components to a test-taker's skill, $p(\Theta|n_i)$ and $p(\Theta|k_i)$ where n_i are the naive, linguistic features associated with item i and k_i are the domain knowledge features associated with item i. In the current sample, we observe that:

$$p(\Theta_i|n_i \cup k_i) < p(\Theta_i|n_i). \tag{2}$$

Since:

$$p(\Theta_i|n_i) + p(\Theta_i|k_i) - p(\Theta_i|n_i \cap k_i). \tag{3}$$

Therefore:

$$p(\Theta_i|k_i) - p(\Theta_i|n_i \cap k_i) < (p(\Theta_i|n_i). \tag{4}$$

To unpack this, we explore the conditional probabilities of each case comparing cases where the student is correct S_c and where the classifier is correct f_c, and where the student is wrong S_w or the classifier is wrong f_w. For the classifier, we have the ground-truth of guessing, but for the students, we can only infer this from the population distribution. We chose a cut-off of two standard deviations above model guessing performance (64%) to select the sub-set of questions on which student population performance was best. These would be classified as "easy" questions by IRT, and performance on them is too high to be explained by naive-feature guessing alone. Additionally we selected questions where student performance was below the chance floor (25%) to serve as our test-set for the model on "hard" questions.

Interestingly, students perform slightly better within the subset of questions correctly guessed by the naive-feature model possibly

Table 3: Student's correct answer rate by model guess rate

Condition	Accuracy
$p(S_c \mid f_c)$	45.2%
$p(S_c \mid f_w)$	42.8%
$p(f_c \mid S_c)$	56.9%
$p(f_c \mid S_w)$	58.2%

suggesting some use of naive-feature guessing by students. However, the model performs above average in both "easy" and "hard" pool of questions.

While this suggests that a naive-features based guessing strategy is compatible with a knowledge-based decision strategy, it also implies that on the hardest questions in this test set, domain knowledge is actually harmful to student performance.

4 ASSESSMENT DESIGN

Resources for best-practice creation of test items exist in the literature [4, 6, 13], these urge care in creation of items such that distractors are plausible, answers are short, there is only one correct answer etc. We wish to add to this list the requirement "cannot be guessed by a machine trained on information that excludes a model of the topic purportedly being tested on." To that end, in this section we describe and deploy a web-based interface that allows instructors and assessment designers to check the likely distribution of guessing based on naive features.

We deployed a system that helps individuals design assessments by using shallow NLP techniques to model novice guessing. Figure 2 shows an application that exposes the classifier via an API and user interface to allow users vet multiple choice questions, surveys, or exams. Using the interface, a user feeds a question and a set of multiple choices to the system. When the **Inkle!** button is clicked the backend API calculates 0-to-1 scores of the choices and returns them to the UI where it displays the scores and draws bar charts based on the algorithm described in the previous sections. In the Figure 2 example, the user enters a question "*Which of the following parts of an animal's body are made of cells?*" and four choices. The correct answer to this question is the third choice "*The brain, but not the muscles.*" and Inklebot correctly estimates that the choice has the highest score of 0.44. The fourth choice receives the lowest score 0.14 even though it has the similar sentence structure and words with the third choice, because of its unique linguistic term "*Neither.*" The remaining two choices mark similar scores 0.22 and 0.2 as their linguistic features are almost identical with each other, except the order of the words.

By observing the calculated scores and reviewing the sentences and the expressions of the questions and the choices she has provided to the system, the user can understand how the linguistic features contribute to the estimation of the correct answer, which can be done by human students in a similar way. At the same time, the Inklebot backend calculates the scores instantly, so that she can freely experiment with variations of the question and the choices that embed different linguistic features. It allows her to interactively edit and improve these questions with the help of the model

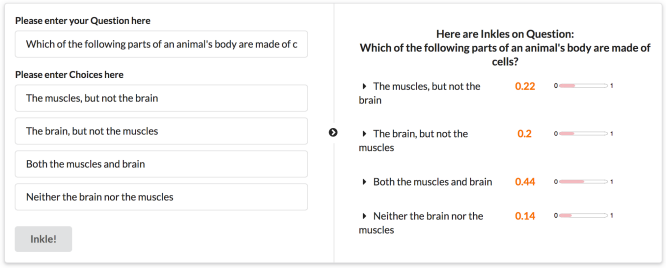

Figure 2: The front-end of "inklebot" the MCQ guesser.

to harden them against guessing strategies that rely on surface features rather than deeper domain knowledge.

5 DISCUSSION & FUTURE WORK

The model reported acts as an expert system for guessing multiple choice questions without knowledge of the question's topic. It can distinguish correct answers slightly better than a large sample students across a wide range of middle- and high school scientific subjects. Further, the distribution of it's guessing behavior provides some insight into the distribution of student responses, and can serve as a rough estimate of how a population of responders might perform on such an exam. While accurate on the AAAS dataset, this accuracy did not extend to the AI2 dataset, suggesting that this later data set relies more heavily on domain-dependent information.

It is somewhat surprising that this guessing behavior does not involve a model of the underlying disciplines nor require factual knowledge in common between subjects. We suggest three possibilities.

First, the model may be able to leverage framing effects [14]. In decision making, framing of a question (i.e., context, word choice, and alternatives) strongly influences users' choices. The features used by the model are just such surface-level linguistic features. Our principle contribution with this technique is to distinguish the relative contributions of framing and domain knowledge to student performance. This is critical information to assess the quality of OER content, and ultimately more accurately estimate and improve student ability. The solution provided here helps independent educators to distinguish between a good guesser, and a knowledgeable learner.

Second, the fact that the model is robust to question difficulty suggests that the "hardest" questions in the data set are in part difficult due to semantic, textual, and logical complexity, rather than domain complexity. Put differently, each item in the AAAS bank is both a science question and an abstract-reasoning question. Items that may require identical levels of science knowledge can be vary substantially in student performance due to variations in the the difficulty of reasoning required to answer the question. Low-knowledge, high-difficulty questions are therefore similar to test items for general fluid intelligence or IQ [3], and assess higher-level, non-domain dependent cognitive abilities.

Finally, there is the possibility of adversarial question generation. Many of the AAAS questions target common misconceptions in student knowledge. In these cases, students factual knowledge

actually hurts their overall performance. The naive features model is immune to such "trick" questions [12].

This model clarifies a possible danger present in OER MCQ content. Not all questions sets are generated for the same purpose. The AAAS set strongly differentiates students on the basis of naive features. This may be desirable in predictive instruments that attempt to estimate student performance across multiple domains. The AI2 test strongly differentiates on factual knowledge, this is desirable in identifying the complexity of domain knowledge mastered by ML models. These sets of assessment materials are specifically adapted to specific contexts and goals, and are not applicable in general. The model presented here helps to support educators who wish to make use of OER assessments by identifying the information upon which the assessments depend.

REFERENCES

[1] 2015. Allen AI2 Science Question Challenge. http://data.allenai.org/ai2-science-questions/. (2015).

[2] François Chollet et al. 2015. Keras. https://github.com/fchollet/keras. (2015).

[3] Andrew R A Conway, Nelson Cowan, Michael F Bunting, David J Therriault, and Scott R B Minkoff. 2002. A latent variable analysis of working memory capacity, short-term memory capacity, processing speed, and general fluid intelligence. *Intelligence* 30, 2 (March 2002), 163–183.

[4] Robert J Dufresne, William J Leonard, and William J Gerace. 2002. Marking sense of students' answers to multiple-choice questions. *Phys. Teach.* 40, 3 (March 2002), 174–180.

[5] Thomas M Haladyna. 2015. *Developing and Validating Multiple-choice Test Items* (3 edition ed.). Routledge.

[6] Moeen-Uz-Zafar Khan and Badr Muhammad Aljarallah. 2011. Evaluation of Modified Essay Questions (MEQ) and Multiple Choice Questions (MCQ) as a tool for Assessing the Cognitive Skills of Undergraduate Medical Students. *Int. J. Health Sci.* 5, 1 (Jan. 2011), 39–43.

[7] Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *International Conference on Machine Learning*. 957–966.

[8] William F McComas. 2014. Benchmarks for Science Literacy. In *The Language of Science Education*. SensePublishers, Rotterdam, 12–12.

[9] Paul McCoubrie. 2004. Improving the fairness of multiple-choice questions: a literature review. *Medical Teacher* 26, 8 (2004), 709–712. https://doi.org/10.1080/01421590400013495 arXiv:http://dx.doi.org/10.1080/01421590400013495 PMID: 15763874.

[10] 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 (2011), 2825–2830.

[11] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.

[12] Dennis M Roberts. 1993. An empirical study on the nature of trick test questions. *Journal of educational measurement* 30, 4 (1993), 331–344.

[13] Michael Rodriguez and Anthony Albano. 2017. *The College Instructor's Guide to Writing Test Items: Measuring Student Learning* (1 edition ed.). Routledge.

[14] Amos Tversky and Daniel Kahneman. 1985. The framing of decisions and the psychology of choice. In *Environmental Impact assessment, technology assessment, and risk analysis*. Springer, 107–129.