Questimator: Generating Knowledge Assessments for Arbitrary Topics

Paper #235

ABSTRACT

Formative assessments allow learners to quickly identify knowledge gaps. In traditional educational settings, expert instructors can create assessments, but it is difficult for novice learners to self assess because they don't know what they don't know. This paper introduces Questimator, an automated system that generates multiple-choice assessment questions for any topic contained within Wikipedia. Given a topic, Questimator traverses the Wikipedia graph to find and rank related topics, and uses article text to form both questions and distractor options. In a study, we found that participants' scores on Questimator-generated quizzes correlated well with their scores on existing online quizzes on topics ranging from philosophy to economics, suggesting that Questimator may be used as an inexpensive formative assessment. More generally, Questimator demonstrates how existing crowdsourced knowledge repositories can be leveraged to support informal and self-directed learning.

Author Keywords

multiple-choice questions, learning, assessment

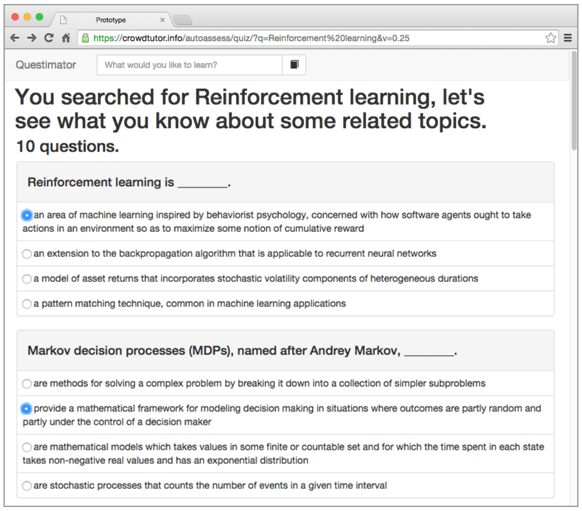
ACM Classification Keywords

H.5.m. Information Interfaces and Presentation (e.g. HCI)

INTRODUCTION

An increasing number of learners are opportunistic and self-driven [6, 39, 8]. In addition, online learning opportunities such as MOOCs attract diverse students with a range of prior knowledge and experience [25]. While online resources such as MOOCs are a boon to self-directed learning, knowledge gaps cause learners to struggle with new material, lose motivation, and even avoid future subjects due to lowered self-perceived efficacy [18].

Formative assessments can help learners diagnose what they do or don't know, so as to direct their learning in more effective ways. For example, a prerequisite to learning about the artificial intelligence topic of "reinforcement learning" is a basic familiarity with probability theory. If a learner does not understand probability distributions, attempting to learn about reinforcement learning would most likely end in frustration. Knowing about their knowledge gap could help them direct their learning to first understand foundational topics before tackling more complex topics.



Prototype
https://crowdtutor.info/autoassess/quiz/?q=Reinforcement%20learning&v=0.25
Questimator
What would you like to learn?

You searched for Reinforcement learning, let's see what you know about some related topics.

10 questions.

Reinforcement learning is _______.

● an area of machine learning inspired by behaviorist psychology, concerned with how software agents ought to take actions in an environment so as to maximize some notion of cumulative reward

○ an extension to the backpropagation algorithm that is applicable to recurrent neural networks

○ a model of asset returns that incorporates stochastic volatility components of heterogeneous durations

○ a pattern matching technique, common in machine learning applications

Markov decision processes (MDPs), named after Andrey Markov, _______.

○ are methods for solving a complex problem by breaking it down into a collection of simpler subproblems

● provide a mathematical framework for modeling decision making in situations where outcomes are partly random and partly under the control of a decision maker

○ are mathematical models which takes values in some finite or countable set and for which the time spent in each state takes non-negative real values and has an exponential distribution

○ are stochastic processes that counts the number of events in a given time interval

Figure 1: Questimator automatically produces quizzes of multiple-choice questions to assess knowledge for arbitrary topics contained within Wikipedia. (The selected answers in the figure are correct answers.)

However, while self-assessment of skills or quality of creative work can be accurate and useful, self-assessment of factual knowledge is more difficult because learners don't know what they don't know [9]. Furthermore, learners tend to overestimate the degree of their factual knowledge, leading to overconfidence [1].

Traditionally, formative assessment of factual knowledge has relied on experts, such as teachers or textbook authors. However, what such expert resources provide in quality, they lack in breadth [13]. Because creating formative assessments of factual knowledge is time-consuming, such assessments are limited to common topics of interest. Furthermore, such assessments are often missing for new topics. How could a self-directed, informal learner assess their knowledge, say, on an arcane but upcoming technology? How could a Star Wars fan gauge their knowledge on the upcoming movie release? Or perhaps more importantly, how could an intelligence analyst track their knowledge of recent geopolitical events?

This paper introduces Questimator, an automated system for generating formative, fact-based assessments on arbitrary topics contained in Wikipedia. A learner uses Questimator by inputting a topic they would like to learn about, and the system generates multiple-choice questions on the target topic and related topics, with each question including a correct an-

Submitted to CHI 2016.

swer and related distractor answers. For example, Figure 1 shows Questimator s automatically generated assessment for the topic of "Reinforcement Learning". Questimator generates questions by leveraging the network structure and content of Wikipedia to find and rank related topics and distractors and to generate question text.

In a controlled experiment with 833 participants from Mechanical Turk, we found that performance on Questimator's generated quizzes are significantly and positively correlated with existing online quizzes across a variety of topics. Overall, students' Questimator-scores correlated with their scores on existing online quizzes (median correlation 0.28). In addition, for certain factually-oriented topics (such as "what is customer satisfaction?"), this correlation was approximately the same as correlations between two existing online quizzes.

Through Questimator, we demonstrate the feasibility of leveraging existing corpora of knowledge such as Wikipedia to create interactive testing materials. Our experience with Questimator suggests that such automated assessments may provide quick, inexpensive formative assessments, leading to more personalized learning experiences and more efficient self-directed learning.

RELATED WORK

Questimator builds from prior approaches for generating assessments by *(i)* automatically creating free-from WH-questions from single documents, *(ii)* automatically creating multiple choice questions using structured lexical databases, and *(iii)* crowdsourcing questions and answers.

Generating WH-Questions from a Single Document

*WH-question*s (*e.g.*, what, where, which, who, how), can be generated a single document, by transforming a declarative sentence into a grammatically correct interrogative sentence [42, 38, 4, 20, 17, 43]. For example, the sentence "Barack Obama is the President of the United States," could be transformed into the question, "Who is the President of the United States?" Given a sentence, choosing a proper question type to ask is non-trivial, and so many systems use an "overgenerate-and-rank" approach, using a supervised machine learning algorithm for ranking, e.g. [17].

The critical problem in this generation process is ranking alternative manipulations of sentences (e.g. the system in [17] achieves only 52% acceptance rate for questions it ranked in its top 20), so additional human intervention is necessary before questions put into use. Systems that have attempted to improve sentence manipulation using, for instance, by using Minimal Recursion Semantics (MRS) to represent the sentences being manipulated [43], have also had limited success.

Furthermore, systems that generate *WH-question*s often simplify the original statement to make transformation easier (e.g. [17]). However, simplification may remove useful details. For example, simplifying "Reinforcement learning is an area of machine learning inspired by behaviorist psychology, concerned with how software agents ought to take actions in an environment so as to maximize some notion of cumulative reward" to "Reinforcement learning is a machine learning method," will result in questions where the learner can no longer distinguish reinforcement learning from other machine learning methods.

Therefore, Questimator restricts its search to direct mentions of the topic, *e.g.*, Barack Obama, and generates questions in a way that does not require transforming the sentence.

The insight Questimator embodies is that it is not always necessary to transform declarative sentences into interrogative ones to generate questions. Instead, Questimator generates "fill in the blank" questions. For example, Questimator might generate the following question, "Barack Obama is _________," and use the remainder of the sentence as the answer, *i.e.* "the President of the United States." This has the additional benefit that sentences do not have to be simplified, preserving all the information from the original sentence.

WH-question approaches are limited to free form responses. That is, learners are expected to type the correct answer to the question, rather than choosing it from among multiple options. This uncued-recall (and how to spell the correct answer) is called for in some applications, but for formative assessment, cued-recall is usually sufficient [35]. Furthermore, Questimator's multiple choice questions can use more involved answers (which may be difficult to type) because test takers do not need to recall the answers but only recognize correct ones. This comes at the cost of Questimator needing to generate appropriate distractors.

Generating Multiple Choice Questions

The generation of multiple-choice questions has been previously explored. Creating a multiple-choice question involves creating three pieces of output: *(i)* a question, *(ii)* a correct answer, and *(ii)* a number of distractors (usually, at least 3). We are primarily concerned with prior systems for generating textual multiple-choice questions, as creating questions for math or logic involves different techniques. Most prior methods for generating text-based questions start with an input article. The question and correct answer is derived from this article (often, but not always, from a single sentence) [19, 7, 30, 26, 41, 40, 2, 16, 31].

Systems differ in how they generate distractors. Some use document-level statistics, *e.g.*, counts of frequency of nouns, and pick related terms (nouns and noun phrases) based solely on that [30]. Others use fixed dictionaries, such as *WordNet*[29] to find synonyms or other related words to use as distractors. WordNet group nouns, verbs, adjectives and adverbs into sets of cognitive synonyms (synsets). It also encodes hyperonymy (super-subordinate relation). Although WordNet has 117,000 synsets, it focuses more on a limited type of cognitive concepts, rather than general knowledge concepts. For example, it contains "baby", and groups it with "infant", but it does not contain "Baby", the Justin Bieber hit song. As compared to more than 4,970,000 Wikipedia articles (by September 2015) for a broad range , WordNet contains relatively small set of concepts, especially for contemporary or popular culture topics. Using terms from the article as dis-

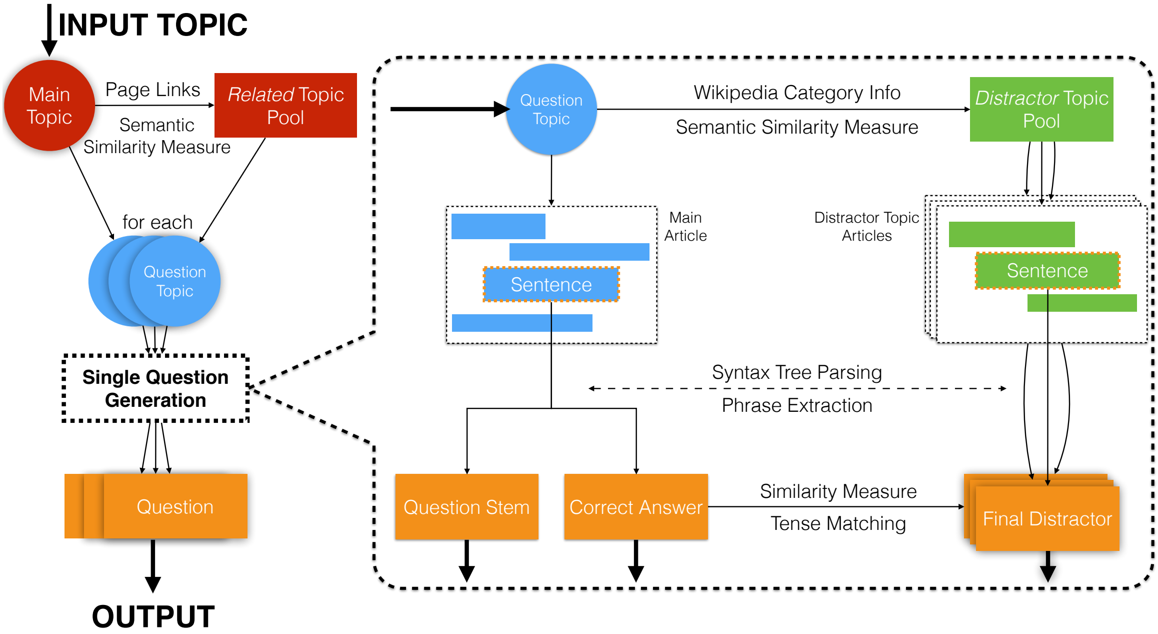


Figure 2: When users enter a topic, Questimator extracts the wiki links on the page and uses similarity measures to find related topics. A question is then generated for both the input topic and each of the related topics. To generate a question from a topic, categories from the Wikipedia page for the topic are used to generate similar topics. Sentences extracted from these pages are then turned into question stems and both correct answers and distractors. In the case of distractors, similarity and tense matching are used to pick the final distractors. The question stem, correct answer, and distractors form the final output question.

tractor can also be problematic, as in many cases, a good distractor might not be frequently mentioned in the article. For example, “Formula Two” is a good distractor for “Formula One”, but does not have a high frequency in the Wikipedia article for “Formula One”. These approaches have therefore mostly been applied to language (vocabulary) learning and assessment, in which differentiating between related words in the language is more relevant to the learning task [19, 7, 30, 26, 40, 16].

Domain-specific ontologies have also been used to generate multiple-choice questions [33, 3, 10]. This approach can be applied to fields other than language learning; however, it requires carefully curated ontology database (also called knolwedge database), which suffers from the same problems as WordNet. For example, [10] deals with only biology questions, and contains 5,500 concepts. Also the relation database contains clear but usually less information about a topic. If it is integrated to text based system, mapping between terms in text and nodes in ontology database can be an issue. Wikipedia has a socially annotated categorical structure[23] and precomputed pagelink information. Questimator treat them as a noisy network structure containing ontology information, and so, unlike these systems, Questimator works across the wide range of domains.

Crowdsourcing Question Generation

Crowdsourcing has recently been explored as a way to generate questions. For example, [11, 12] ask crowdworkers to directly generate the questions by mining existing Q&A sites. For example, Stack Overflow can be used as a scalable source of coding questions. In fact, a simple web search reveals a number of existing quizzes online for a variety of topics. These are sometimes posted by instructors, text book writers, online classes, *e.g.*, MOOCs, or even students. A challenge with these approaches is verifying that the questions are of high quality. Although Wikipedia is crowdsourced, popular articles are quite accurate and complete, and so the questions generated by Questimator are in a sense crowdsourced from source material subject to Wikipedia’s quality control.

Questimator represents a new approach to multiple-choice question generation. Inspired by the recent trend to utilize large scale datasets, and enabled by the corresponding development in NLP, Questimator uses a connected set of documents rather than a single document to generate multiple choice questions for general topics.

Questimator

Questimator is our system for automatically generating multiple choice questions from Wikipedia. To use Questimator, users first enter a topic they would like to know about (the interface helps match the entered topic to one that appears in Wikipedia). Questimator then returns a set of multiple choice questions automatically generated from Wikipedia chosen to assess the user’s knowledge of the provided topic (Figure 1 shows the first two questions generated for the topic “reinforcement learning”). The number of questions returned is configurable, although by default Questimator returns 10 questions. To do this, Questimator:

1. generates and ranks a list of related topics,
2. generates questions for each of the topics by
   (a) generating question stems for each,
   (b) generating and ranking distractors for each question.

STAGE 1: RELATED TOPIC IDENTIFICATION

The input of Questimator is a single topic, *e.g.* "reinforcement learning" or "France." Questions to assess knowledge of that topic need to be broader than asking about the specific input or main topic itself. The way Questimator approaches this is to find topics that are related to the main topic and generating questions for them as well.

Related topics can be drawn from prerequisites, subtopics and parallel topics. To generate candidate related topics, Questimator takes the union of the wikilinks on the main topic page, the category links, and the back links (topics that point to the main topic page). Questimator thus leverages both the Wikipedia document for the main topic, and the larger Wikipedia structure to generate and rank related topics.

To rank these candidate related topics, Questimator uses three measures of the similarity:

- number of times the term appears in the main topic article
- backlink overlap of the candidate topic with the main topic
- semantic similarity between topics

Questimator combines these measures to produce a single relatedness metric, by sum after normalization. And we then we rank topics based on that, choose the top $n (n \geq 3)$ items as related topics to generate questions for the quiz. We will elaborate each of the three features in the following section.

Wikilink term frequency

Important concepts are likely to be repeatedly mentioned in the *main article*, and so term frequency can be an important feature for identifying important subtopics. We first extract all the links to other Wikipedia topics in the main article (called *pagelinks* or *wikilinks*). And given this candidate pool of the related topics, we count the term frequencies of the wikilinks in the main article itself. We take simple correferences into account, like explicitly stated abbreviation and the exact text in the article linked to the corresponding wikilinks. For example, "MDP" for "Markov Decision Process" given "Markov Decision Process (MDP)", and "behaviorist psychology" for "Behaviorism" given it links to "Behaviorism" in the main article. The term frequencies alone do not always yield good related topics. For example, "(software) agent" are mentioned many times in the article for "Reinforcement learning", but it is not an important related concept. Neither is "Real number" for "Invertible matrix".

Backlink Overlap

Through observation, the topics brought up together with the main topic frequently are very likely to be realted. Thus we take backlinks to a topic, i.e. articles of which topics mention the topic, into account. We use the pagelink dataset dumped from Wikipedia. This dataset contains information about which topic links to which. Similar backlink coverage may infer topic relation. For the topics in the candidate pool, we find the backlink overlaps between the candidate topics and the main topic, and compute the similarity between their backlinks using the following formula (cosine similarity):

$$\text{sim}(A, M) = \frac{|\text{backlinks}(A) \cap \text{backlinks}(M)|}{|\text{backlinks}(A)| \cdot |\text{backlinks}(M)|}$$

, where $|\cdot|$ is the size of the set, M is the main topic, A is any candiate topic. Basically the higher percentage of backlinks a topic and the main topic share, the more similart they are.

This allows us to identify topics that are overlooked in the main article that can not be identified by term frequency. For example, "Convolutional neural network" for "Deep learning", and "Affine transformation" for "Rigid transformation".

Embedded semantic similarity of topics

We used a Word2Vec[28] model trained on Wikipedia articles with topics (article titles) treated as independent entities. This produces a vector representation for each title and word. We then measure the similarity between topics by their vectors' cosine similarity. If we cannot retrieve a vector for a title as a whole, we use the average the vectors for each word.

Word2Vec provides vector reprensentations for words and can be easily generalized to preidentified noun phrases and other entities in text. Semantically similar words and entities are mapped to similar vectors. We use a Word2Vec[28] model trained on Wikipedia articles with wikilinks treated as independent entities. This way, we have a direct and supposedly more precise measurement of the semantic similarites between the topics. The semantic similarity of two wikilinks is simply the cosine distance of the vector representations for the topic titles. Due to various reasons, the Word2Vec model may not have an entity vector for every article title. For example, the dumped wikipedia article set the Wor2Vec was trained on is not totally the same from the current one, terms may be changed or missing. In this case, we will treat the title of the wikilink as a list of words, and retrieve from the model a vector for each of the word, and average the word vectors to get a vector for the title. We also use the same embedding method for semantical similarity measure for finding candidate distractor topics later when we generate single questions. Word2Vec outperforms WordNet [29] to find similar terms in most, if not nearly all, cases.

Adding this feature to the relatedness measure, we might pick out "Backpropagation" for "Artificial neural network", which does not have either high term frequency or backlink overlap, but is an important optimization method for neural networks.

Combine and Rank Relatedness

We combine the above three perspectives, and rank accordingly. For each of the three features X above, we normalize it by:$X' = \frac{X - \text{mean}(X)}{\sqrt{\text{variance}(X)}}$. Then X' is truncated to be inside $[-1, 1]$, so a single feature would not play a too significant role. We then sum the three normalized features to create a combined relatedness measure. Each link is then sorted by its relatedness value, and the top topics are used as related topics to generate questions for the main topic.

STAGE 2: SINGLE QUESTION GENERATION

A multiple choice question (MCQ) is composed of *question stem*, a single *correct answer* and a set of *distractors*. A question stem is the part where the item to ask is stated, for example "Reinforcement learning is ______.". We generate *cloze questions*, namely *gap-fill questions*, as the stems. Distractors are the other options shown together with the correct answer, among which the examinee have to choose from.

The single question generation part of Questimator, take a topic as input, and generate a single MCQ:

- Question stem and correct answer are generated first from the same sentence, and
- distractors are generated later separately from articles of similar topics.

We elaborate on the above two phases of the pipeline separately in the following section.

STAGE 2(A): QUESTION STEM GENERATION

For a given topic, Questimator retrieves the Wikipedia article of the topic, and generates a question stem by

- finding a set of sentences that each contain the stemmed tokens of the topic string,
- parsing a sentence into a syntax tree, and
- if the syntax tree matches certain patterns, extracting a question stem and the correct answer.

Sentences Selection

We segregate sentences, and use regular expression to find sentences containing stemmed tokens of the topic string. Then Questimator processes the sentences by their natural order, as the first few sentences where the main topic is mentioned usually contain an explanation of that topic.

Parsing and Phrase Extraction

Questimator uses PCFG (Probabilistic Context Free Grammar) syntax trees to extract question stems. Stanford parser [24] is used to parse the sentences into PCFG syntax trees. Then Questimator uses TGrep [37] expression to do pattern matching to extract desired phrases we want to ask about. If matched, we substitute the matched phrase with blank to generate the question stem, and that phrase *per se* will be the correcet answer. We then stop and proceed to distractor generation.

Questimator generates questions asking about the topic noun phrases in the main clause or the verbal phrases after them, as directly testing on concept explanation are frequently used in expert assessments. Additional question types [17] can be generated by simply inserting the TGrep patterns into it. We contribute to complete the graph of question generation, focusing on related topic search and distractor generating from multiple documents which were less explored by former literatures. As a byproduct of limiting the question types, we avoid the problem of choosing the proper question type.

Given a definition sentence of a topic, for example, "Reinforcement learning is a machine learning method that …". We may generate question stem like "______ is a machine learning method that …" and "Reinforcement learning is ______". As the input topic to the Questimator is *Reinforcement learning*, the first question can be easily guessed to be "Reinforcement learning". So we incline to generate question asking the verb phrases aforementioned. The evaluation of the questions later is based on this type.

STAGE 2(B): DISTRACTOR GENERATION

Given the question stem and correct answer, we are a few distractors away from getting a complete MCQ.

To generate distractor, Questimator

- finds topics in the same categories as the main topic,
- ranks them by their Word2Vec semantic similarities with the main topic, and picks the top few topics,
- extracts one distractor phrase for each, using the same process of extracting the right answer earlier,
- uses skip-thought to find distractor phrases most similar to the right answer phrase.

Distractor Topic Selection

Questimator first finds topics to construct a candidate pool using category information, and use semantic similarity to pick distractor topics from them. If due to various reason, the semantic similarity measurement fails, we fall back to use the rank of sharing category counts to select distractor topics.

Unlike searching for related topics to generate a coherent quiz for the main topic, the criterion of similarity changes here. Relatedness is still important, but not the single most important factor. For example, "Camera Matrix" is a very related topic for "Camera Calibration", but as a distractor, examinees can easily tell a matrix from a calibration process. Parallel topics that are hard to distinguish are preferred. For example, "Supervised learning" for "Unsupervised learning", "Forumula One" and "Formula Two".

Category information

Questimator finds category information by using Wikipedia's socially annotated category hierarchy. We rank topics by the count of sharing categories between a candidate topic and the main topic. However, category sharing alone will not get expected distractors. Topics remotely related might be from a same category, for example, "Newton's laws of motion" and "Catherine Barton" (Issac Newton's half niece) are in the same category "Isaac Newton".

Embedded semantic similarity of topics

The topic semantic similarity measure is the same as the measure for related topic explained in the former section .

Questimator ranks the distractor topics sharing at least one cateogy with the main topic by their Word2Vec similarity with the main topic, and select the top ones. For n ($n = 3$ by default) distractors, Questimator intermediately select $m \times n$ distractor topics ($m = 3$ by default) to generate the distractor phrases. During the distractor phrase extracting process, Questimator throws away phrases containing the stemmed tokens of the main topic string. Because there is a high probability that this will reveal itself as a wrong answer.

Embedded similarity of distractor phrases

After we picked the distractor topics, and generated distractor phrases from them, we want to measure the similarity between the distractors and the correct answer, and pick the ones most difficult to distinguish from the correct answers. Questimator applies skip-thought vectors [21].

Skip-thought is a principled way to embed sentences into vector space. Like Word2Vec generating vectors for words, skip-thought generates vectors for sentences, with semantically similar sentences having similar vectors. Questimator uses a pre-trained model of skip-thought vectors. The model is trained on 11,038 novels in 16 genres [21], and skip-thought used pre-trained Word2Vec models to linearly map words and extend the vocabulary. Coupled with the distractor topic similarity measure, it gives us a similarity measure that helps better filter distractors that can be easily identified as a wrong answer.

After the raw distractors are extracted from the articles, we match their tense with the correct answer. This prevents the examinees from identifying them as wrong answers simply by spotting tense mismatch.

Finally distractors together with the question stem and the correct answer are delivered as a whole MCQ.

EVALUATION

The goal of Questimator is to generate good MCQ quizzes for knowledge evaluation for arbitrary topics. We focus on two key assessment measures of Questimator:

1. the correlation between a student's performance on a set of Questimator automatically generated questions, and the same student's performance on a set of human expert generated questions, and

2. the discriminatory power of individual Questimator-generated questions, in terms of their ability to distinguish between students' with varying topic knoweledge states.

Our motivation for the first objective is that, we would like Questimator to automatically construct a quiz that provides a measure of student knowledge that is similar to an assessment constructed by expert teachers. If Questimator can provide assessments such that a student performance on said assessments correlates highly with the same student's performance on an expert-constructed assessment, then that provides encouraging evidence that Questimator is able to capture signals that provide important insight into a student's state of knowledge.[1]. Therefore, we focus our evaluation on comparing student performance on quizzes generated by Questimator to those generated by experts on a diverse set of topics.

An alternative evaluation strategy could examine how frequently Questimator questions exhibit desirable attributes in questions, such as relevance, question type, syntactic correctness and fluency, or ambiguity. This is the approach used by [17, 43]. While this evaluation measures how similar machine-generated questions and expert-generated questions *look*, it does not assess the actual ability of questions to discriminate between different levels of knowledge. For example, while both expert-generated and machine-generated questions might be ambiguous, expert-generated questions might nudge students to think more critically, while machine generated ones might not. To enhance ecological validity, Questimator assesses actual student performance and compares this across automatically and expert-generated quizzes.

The second objective stems from wanting a deeper understanding of the quality of the individual questions generated by Questimator. We would like to better quantify how effective different automatically-constructed items are at assessing student knowledge, and how these compare to expert-generated questions. This could also have interesting implications for future work which may wish to generate many questions and then subsample.

Experimental Setup

Our evaluation was a within-subjects experiment with participants drawn from Amazon Mechanical Turk. In this setup, each learner sees a quiz on a particular topic that is composed of both expert and Questimator generated questions.

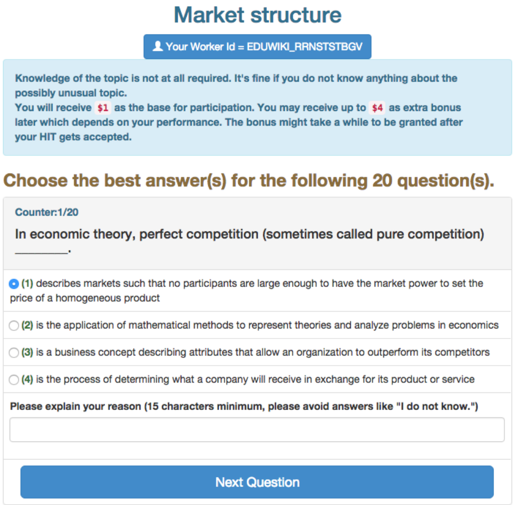
Choosing topics We evaluated Questimator on 10 diverse topics. We chose topics for evaluation based on three criteria. First, we chose topics of broad interest because our participants were drawn from Mechanical Turk, which excluded topics like "Reinforcement Learning" that we thought few workers would know about. Second, we chose topics for which we could find existing online quizzes in order to compare to human expert-generated assessments. Finally, we chose topics that naturally lent themselves to textual questions and answers because Questimator does not yet handle mathematical symbols, images, or video. With these criteria, we chose ten topics across ten disciplines [Table 1]. Quizzes were drawn from MOOCs (hosted by Coursera/edX), US university/school board websites, and textbooks by major publishers (*e.g.*, McGraw Hill). We identified two expert-generated quizzes for two topics, and one for the other topics (Table 1).

Selecting questions Expert-generated quizzes varied in their length from 10 to 60 questions. To eliminate testing differences across topics, we randomly sampled 10 questions from each expert quiz, after removing questions that relied on numerical calculations and True-False questions wherever possible (if removing these questions resulted in fewer than 10 questions, we retained them). One of the two expert-generated quiz on the "Vietnam war" was drawn from a textbook chapter on the "Vietnam Era", and included questions on contemporary issues like the Civil Rights Movement. For uniformity, we removed these questions before sampling. For each topic, we also generated 10 questions on Questimator.

[1] Indeed, correlation alone may be sufficient to create a system that can be used for certification. On the other hand, high correlation between an automatically constructed assessment may be useful, but not sufficient for identifying the important aspects to teach a student. For example, imagine that a student's ability to define a particular technical term is strongly correlated with his/her knowledge of a particular algorithm. Asking a student to memorize that technical term would improve his/her performance on the assessment without increasing his/her understanding of how the algorithm works or how to implement it.

Discipline	Questimator Query Term	Expert quiz source (Chapter/quiz title)
Business and marketing	Customer satisfaction	Coursera (Customer Centricity)
Earth science	Earthquake	McGraw Hill (Earthquakes) stjames.k12.mn.us (Earthquakes)
Psychology	Developmental psychology	McGraw Hill (Human Development)
Biology	Cell (biology)	McGraw Hill (Cell Structure and Function)
Economics	Market structure	McGraw Hill (Market structure and Imperfect competition)
History	Vietnam War	uco.edu (Vietnam Era) softschools.com (Vietnam War)
Philosophy	Metaphysics	McGraw Hill (Introducing Metaphysics)
Medicine	Stroke	emedicinehealth.com (Stroke)
Environmental Science	Waste management	McGraw Hill (Solid Waste Management and Disposal)
Physics	Elasticity (physics)	McGraw Hill (Elasticity)

Table 1: Expert Quiz Sources.



Market structure

Your Worker Id = EDUWIKI_RRNSTSTBGV

Knowledge of the topic is not at all required. It's fine if you do not know anything about the possibly unusual topic.
You will receive $1 as the base for participation. You may receive up to $4 as extra bonus later which depends on your performance. The bonus might take a while to be granted after your HIT gets accepted.

Choose the best answer(s) for the following 20 question(s).

Counter:1/20

In economic theory, perfect competition (sometimes called pure competition) ________.

● (1) describes markets such that no participants are large enough to have the market power to set the price of a homogeneous product

○ (2) is the application of mathematical methods to represent theories and analyze problems in economics

○ (3) is a business concept describing attributes that allow an organization to outperform its competitors

○ (4) is the process of determining what a company will receive in exchange for its product or service

Please explain your reason (15 characters minimum, please avoid answers like "I do not know.")

Figure 3: Evaluation Experiment Interface. Crowdworkers are asked to answer mingled expert/Questimator questions one by one.

Creating quizzes To generate a quiz for a topic, we combined 10 questions from an expert quiz[2] with the 10 Questimator questions. For two topics for which we had two expert quizzes, we generated a quiz which comprised 20 expert questions only, as sampled from both quizzes.

Participants

Participants were recruited from Mechanical Turk. In all, 833 workers participated. All participants were paid $1 as base for their participation. In addition, they could earn up to $4 as bonuses based on their test score, as described below.

Experimental procedure

Participants were shown one question at a time (Figure 3). To reduce ordering effects [27], question order was randomized across participants. To reduce response-order biases [5], the order of answer choices was also randomized.

To encourage participants to put in their best effort, we incentivized performance with a bonus payment of up to $4. Each question was also augmented with a textbox that asked participants to to explain their reasoning (at least 15 characters), a technique that has been previously shown to encourage honest effort [22]. Finally, to discourage participants from using the Internet to search for answers, we monitored the web browser `blur` event and warned subjects that they would not be allowed to submit an answer if they left the window. Subjects spent an average of 58 seconds on each question. The average length of textbox comments was 66 characters, much longer than the required 15 characters. We received between 78 and 82 completed quizzes for each topic.

Quiz	Correlation
Customer satisfaction	0.4651**
Earthquake	0.3698**
Developmental psychology	0.3665**
Cell (biology)	0.3364**
Market structure	0.2823*
Vietnam War	0.2750*
Metaphysics	0.2585*
Stroke	0.2551*
Waste management	0.2163
Elasticity (physics)	0.0830

Table 2: Pearson correlations of quiz scores, ordered by correlation values. (**:p-value < 0.01, *:p-value < 0.05)

Objective 1: Correlation with Expert-Generated Quizzes

We computed a score for both the expert- and Questimator-generated questions for each participant. Each question was weighted equally. The median Pearson correlation between Questimator and expert quiz scores was 0.28 (Table 2). The highest correlation was 0.47 for "Customer satisfaction", and the lowest was 0.08 for "Elasticity (physics)".

Of course, even expert-generated quizzes may not correlate highly: two experts may focus on different subtopics within a general topics, or prioritize different forms of knowledge. Therefore, we also evaluated how well scores on two expert quizzes correlate for two topics. The Pearson correlation of the two scores for expert quizzes was 0.4304 for the Earthquake topic (vs 0.3698 for the Questimator-expert) and 0.4595 for "Vietnam War" (vs 0.2750 for the Questimator-

[2] If two expert quizes were available, we selected one at random

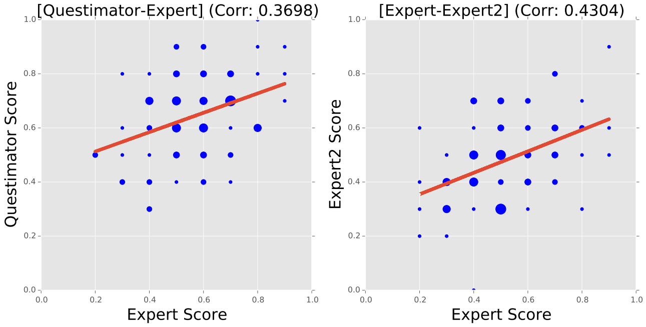


Figure 4: Scatter plots of scores (Expert-Questimator and Expert-Expert comparison for “Earthquake”). X-axis is the score for the reference (expert) quiz, Y-axis is score for Questimator quiz or another expert’s quiz. Blue nodes: score, radiuses are in portion to the points overlapping at the that location, with the smallest node in graph corresponding to 1 point. Red line: linear regression of scores.

expert). Figure 4 shows the scores and correlations between Questimator-expert and expert-expert for “Earthquake”, to illustrate the variability and correlation between different groups of assessment questions.

Taken together, these results suggest that Questimator scores generally correlate quite well with expert quiz scores, and at least for one topic (Customer Satisfaction), this correlation value is as high as the correlation between two expert quizzes.

Objective 2: Discriminative Power of Questions

We also wanted to analyze the quality of the individual questions generated by Questimator. Such an analysis both will yield further insight into its performance, and may be useful for future versions of the system.

To do so, we used a very popular approach from psychometrics, Item Response Theory (IRT) which is used to evaluate test items and analyze test takers. In IRT, each question is called an *item*. We fit a two-stage IRT model to analyze questions from a quiz under investigation (Questimator quiz) against an gold standard quiz (expert quiz) when they are mixed together for testing.

The IRT model we use is the unidimensional dichotomous model. Given the student ability $\theta \in \mathbb{R}$, the probability of success on the jth question item is

$$P(Y_j = 1|\theta) = \frac{1}{1 + e^{-\alpha_j \theta + \beta_j}}, \qquad (1)$$

where θ is the student ability, α_j and β_j are question parameters. θ is a scalar, implying single knowledge ability for the related term affects students’ performance on the quiz. Naturally α is also unidimensional. We selected this simple version of an IRT model to reduce overfitting, and because it has the benefit of being easily interpretable and visualized. α_j is the question discriminative parameter, and β_j is question difficulty. In the IRT curves of of an item (in our case, as question), α specifies the curve steepness (larger α, steeper curve) and β shifts the curve horizontablly (larger β, more to the right). More specifically, we considered a Bayesian IRT model with a prior of $\theta_i \sim N(0, 1)$, $\alpha \sim N(\mu_\alpha = 1, \delta_\alpha^2)$, $\beta \sim N(\mu_\beta = 0, \delta_\beta^2)$. m_α is set to be larger than 0 as we would suppose the questions have some positive discrimination effect.μ_α, μ_β, δ_α and δ_β can be set to reflect prior knowledge of the question items. We use MCMC to perform Bayesian inference and estimate the parameters of the model. We sample for 20000 iterations, and burn the first 15000 iterations. Please refer to [15] for further details.

Quiz	Expert $\bar{\alpha}$	Questimator $\bar{\alpha}$
Customer satisfaction	0.40284	0.81028
Market structure	0.45128	0.56934
Earthquake	0.55706	0.56066
Cell (biology)	0.78982	0.55746
Waste management	0.62222	0.47502
Metaphysics	0.6299	0.47246
Stroke	0.26404	0.47116
Developmental psychology	0.93632	0.46008
Vietnam War	0.83662	0.35828
Elasticity (physics)	0.4334	0.3247
Average	0.6421	0.5248

Table 3: Mean of the individual question discrimination α parameters for each quiz.

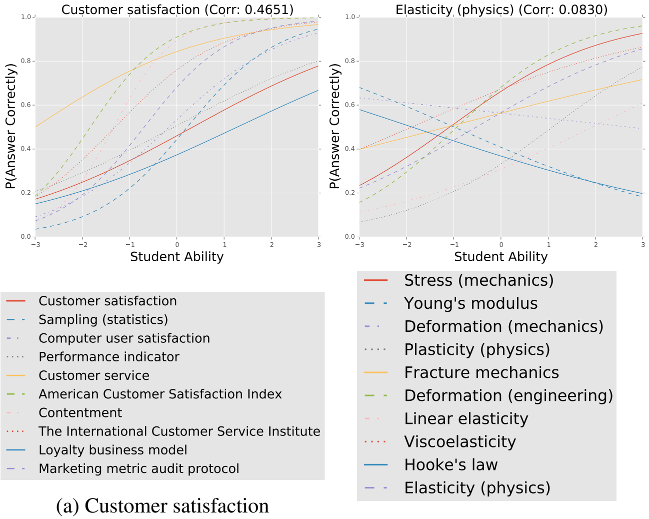
We have two types of questions (from experts and from Questimator) and if we estimated the IRT parameters independently, it would be less clear how to compare them. Instead, we first fit the expert questions with the above Bayesian IRT model, and use these to get an estimation of a student’s ability parameter $\hat{\theta}_{exp}$s. We then treat these $\hat{\theta}_{exp}$s as the true underlying student abilities of the tested topic, and estimate only α and β for the Questimator questions. Figure 5 shows some example resulting IRT curves for two topics.

For “Customer satisfaction” (Pearson correlation across overall expert-Questimator question sets: 0.4651), all of the questions has positive discrimination ($\alpha > 0$), and a fair amount have relatively large αs (steep sigmoid curve). On the other hand, for “Elasticity (physics)” (correlation: 0.0830), the questions have lower discriminative power (smaller αs). Indeed, a few have negative discrimination coefficients ($\alpha < 0$), meaning the more knowledgeable of Elasticity, the less likely student is going to get them correct.

Table 3 shows the average α parameters values across the questions for each topic, both for the existing online quizzes and our Questimator-generated quizzes. These results suggest that for many topics Questimator is identifying questions that have positive discriminative power. In the future, it would be interesting to use this approach to automatically refine generated questions or question generation techniques.

DISCUSSION

One of our goals in developing Questimator is for it to generate multiple-choice quizzes at scale for topics that do not already have quizzes available online.



(a) Customer satisfaction

(b) Elasticity (physics)

Figure 5: IRT curves of Questimator questions. Each curve corresponds to a question. Related topics used to generate each question are shown in the legend.

Generate questions on niche and contemporary topics

Figure 6 shows example questions generated for three topics. These topics were drawn from the list of most popular Wikipedia articles (We maintain a constantly-updated corpus of such quizzes at **https://crowdtutor.info/autoassess/quiz_list/**). For many topics, such as "Miley Cyrus", "Wonders of the World", or "Oasis (band)", no existing quizzes are readily available. For other topics like the ongoing "Syrian Civil War", expert-generated quizzes, even if available, will become quickly out-of-date. Finally, for some topics, such as "Fascism" or "Constructivism (philosophy of education)", we found that Questimator generated questions similar to those on existing quizzes.

Limitations in the current Questimator system

We manually inspected multiple-choice quizzes on more than a hundred topics, and found three common error-modes.

Because Questimator uses the target article's text as the starting point for generating questions, we find its ability to generate good questions depends critically on the quality of the article text itself. First, when when the target article was too short (e.g. a Wikipedia "stub"), Questimator was unable to find enough related terms to generate a coherent quiz. Second, the current implementation of Questimator does not handle co-references well. For example, an article about "Ocean gyre" (a large system of circular ocean currents) may shorten it to "gyre" in most of the article, preventing Questimator from finding sentences about "ocean gyres."

Questimator also fails when it can't find enough information to distinguish the topic from related topics. For example, the exact phrase ”an American singer, songwriter and actress” is used in Wikipedia to describe many artists, like Miley Cyrus, Katy Perry and Taylor Swift, leading to questions that test arcane knowledge (*e.g.*, Figure 6, which asks if Miley Cyrus is also a record producer and choreographer).

Finally, Questimator does not reason about the real world. For example, it does not understand that "Syria Civil War" cannot be a military conflict in Lebanon (being a civil war).

Test taking strategies

Recall that in addition to marking the right answers, participants were also to provide a text explanation of their reasoning. While most responses were not informative ("It just sounds right to me."), we found some participants guessed based on their knowledge of related topics. For example, given a question on President Nixon's action in the Vietnam War, one participant remarked, "while I wasn't alive when he was president, Nixon doesn't strike me as a peaceful guy, so those answers were out." Similarly, other participants used their general knowledge of the business world to answer the Customer Satisfaction quiz: "Customer centricity is the main factor to determine the [success of] business." It is likely that Questimator questions are especially susceptible to such test-taking strategies as they use knowledge of related topics to generate questions.

Promising approaches that were not effective

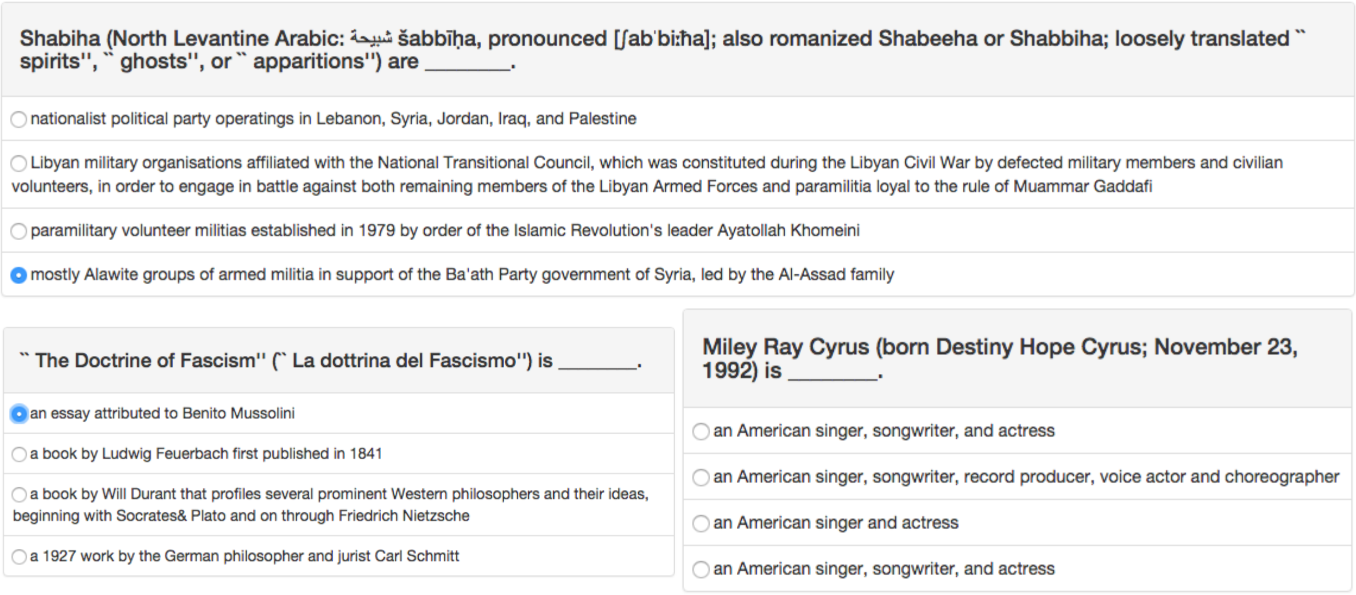
Because Wikipedia has a consistent editorial style (especially for popular or featured article), simple approaches to selecting sentences stems work surprisingly well. In particular, Questimator processes sentences in order, as the first few sentences often explain the main topic. We also tried more complicated approaches, like LexRank [14], a popular document summarization method, but found it gave worse results.

To cover the main concepts of a topic comprehensively, related topics should ideally be maximally diverse. However, when we tried to use clustering techniques, like k-means and DBSCAN, results were dissapointing. Possibly due to the relative link sparsity, we found these approaches usually resulted in one big cluster with majority of the candidate topics, and other clusters which contained single topics. Given these limitations, the current implementation of Questimator does not optimize the diversity of related topics.

What are Questimator-quizzes suited for?

The goal of Questimator is to generate formative, fact-recall assessments for a broad range of topics. Our empirical evaluation found that Questimator-scores correlate significantly with scores on existing quizzes on a broad variety of topics, suggesting utility as formative assessment. For a self-directed learner, early, even approximate feedback about their gaps in understanding is useful in guiding future learning [32].

As such, Questimator quizzes are not designed as a replacement for expert-generated quizzes. Still, our empirical evaluation of Questimator shows that for many question types, Questimator scores correlate well with existing quizzes, and for some topics, score-correlations with an expert quiz and are as high as correlations between two expert quizzes themselves. This suggests that approaches similar to Questimator may also help *summative* assessments, which can be used for applications such as certification. Assessments that go beyond fact-recall, and test deductive thinking, as well as questions that elicit an open-ended response may be necessary.



Shabiha (North Levantine Arabic: شبيحة šabbīḥa, pronounced [ʃabˈbiːħa]; also romanized Shabeeha or Shabbiha; loosely translated `` spirits'', `` ghosts'', or `` apparitions'') are ________.

○ nationalist political party operatings in Lebanon, Syria, Jordan, Iraq, and Palestine

○ Libyan military organisations affiliated with the National Transitional Council, which was constituted during the Libyan Civil War by defected military members and civilian volunteers, in order to engage in battle against both remaining members of the Libyan Armed Forces and paramilitia loyal to the rule of Muammar Gaddafi

○ paramilitary volunteer militias established in 1979 by order of the Islamic Revolution's leader Ayatollah Khomeini

● mostly Alawite groups of armed militia in support of the Ba'ath Party government of Syria, led by the Al-Assad family

`` The Doctrine of Fascism'' (`` La dottrina del Fascismo'') is ________.

● an essay attributed to Benito Mussolini

○ a book by Ludwig Feuerbach first published in 1841

○ a book by Will Durant that profiles several prominent Western philosophers and their ideas, beginning with Socrates& Plato and on through Friedrich Nietzsche

○ a 1927 work by the German philosopher and jurist Carl Schmitt

Miley Ray Cyrus (born Destiny Hope Cyrus; November 23, 1992) is ________.

○ an American singer, songwriter, and actress

○ an American singer, songwriter, record producer, voice actor and choreographer

○ an American singer and actress

○ an American singer, songwriter, and actress

Figure 6: Examples of the questions generated. Corresponding to "Shabiha"(for quiz "Syrian Civil War"), "The Doctrine of Fascism"(for quiz "Fascism"), "Miley Cyrus"(for quiz "Miley Cyrus").

CONCLUSIONS AND FUTURE WORK

This paper has introduced Questimator, a system for generating formative, fact-recall questions on arbitrary topics. Our results show that our automatically generated questions are comparable to existing online quizzes on a variety of topics. Future work will explore improving the quality and the breadth of automatically created assessments.

A broader variety of questions

Questimator assesses fact-recall through multiple-choice questions. Could we generate other kinds of assessment? For example, could a similar network-based model be used to allow students to complete a sentence fragments by filling in a blank? Furthermore, Questimator currently only asks questions of identity ("X is Y"). Future work could also examine questions of entailment ("X means Y") or causality ("X causes Y"), which are more important in different disciplines.

Using crowds to improve question quality

Could crowds continually improve the quality of questions Questimator generates? Using an an active learning framework (and the IRT analysis described earlier), future work could preserve the breadth of Questimator for infrequently accessed topics, and approach the quality of expert-generated quizzes for topics that are more popular.

Learning through testing

Testing students on knowledge they are about to gain (pretesting) can enhance learning [36, 34]. With Questimator, independent learners learning arbitrary topics can easily access MCQ quizzes for those topics. In future work, these questions could be integrated into personalized learning systems to leverage the testing effect. For example, a "learning interface" to Wikipedia could show readers questions *before* they read an article. Furthermore, such fact-recall tests could personalize the content of knowledge sources themselves. Continuing with our imagined Wikipedia interface, future work could expand or summarize parts of articles based on what readers already know. We can assess the background knowledge of different aspects of the topic based on the testing results, and tailor the content we show to the learners.

REFERENCES

1. 2002. Webbased Learning: Sound Educational Method or Hype? A Review of the Evaluation Literature. *Academic Medicine* 77, 10 (October 2002), S86–S93.

2. Manish Agarwal and Prashanth Mannem. 2011. Automatic gap-fill question generation from text books. In *Proceedings of the 6th Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics, 56–64.

3. Maha Al-Yahya. 2011. OntoQue: a question generation engine for educational assesment based on domain ontologies. In *Advanced Learning Technologies (ICALT), 2011 11th IEEE International Conference on*. IEEE, 393–395.

4. Husam Ali, Yllias Chali, and Sadid A Hasan. 2010. Automation of question generation from sentences. In *Proceedings of QG2010: The Third Workshop on Question Generation*. 58–67.

5. Niels J Blunch. 1984. Position bias in multiple-choice questions. *Journal of Marketing Research* (1984), 216–220.

6. Joel Brandt, Philip J Guo, Joel Lewenstein, Mira Dontcheva, and Scott R Klemmer. 2009. Two studies of opportunistic programming: interleaving web foraging, learning, and writing code. In *Proceedings of the*

SIGCHI Conference on Human Factors in Computing Systems. ACM, 1589–1598.

7. Jonathan C Brown, Gwen A Frishkoff, and Maxine Eskenazi. 2005. Automatic question generation for vocabulary assessment. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 819–826.

8. Carrie J Cai, Philip J Guo, James R Glass, and Robert C Miller. 2015. Wait-Learning: Leveraging wait time for second language education. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 3701–3710.

9. Deanna Caputo and David Dunning. 2005. What you dont know: The role played by errors of omission in imperfect self-assessments. *Journal of Experimental Social Psychology* 41, 5 (2005), 488–505.

10. Vinay K Chaudhri, Peter E Clark, Adam Overholtzer, and Aaron Spaulding. 2014. Question generation from a knowledge base. In *Knowledge Engineering and Knowledge Management*. Springer, 54–65.

11. Maria Christoforaki and Panagiotis Ipeirotis. 2014. STEP: A Scalable Testing and Evaluation Platform. In *Second AAAI Conference on Human Computation and Crowdsourcing*.

12. Maria Christoforaki and Panagiotis G Ipeirotis. 2015. A system for scalable and reliable technical-skill testing in online labor markets. *Computer Networks* (2015).

13. Jannette Elwood and Val Klenowski. 2002. Creating communities of shared practice: the challenges of assessment use in learning and teaching. *Assessment & Evaluation in Higher Education* 27, 3 (2002), 243–256.

14. Günes Erkan and Dragomir R Radev. 2004. LexRank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research* (2004), 457–479.

15. Jean-Paul Fox. 2010. *Bayesian item response modeling: Theory and applications*. Springer Science & Business Media.

16. Donna Marie Gates, Greg Aist, Jack Mostow, Margaret McKeown, and Juliet Bey. 2011. How to Generate Cloze Questions from Definitions: A Syntactic Approach. In *2011 AAAI Fall Symposium Series*.

17. Michael Heilman and Noah A Smith. 2010. Good question! statistical ranking for question generation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 609–617.

18. Horwitz M.B. Horwitz, E.K. and J. Cope. 1986. Foreign language classroom anxiety. *The Modern language journal* 70, 2 (1986), 125–132.

19. Ayako Hoshino and Hiroshi Nakagawa. 2005. A Real-time Multiple-choice Question Generation for Language Testing: A Preliminary Study. In *Proceedings of the Second Workshop on Building Educational Applications Using NLP (EdAppsNLP 05)*. 17–20.

20. Saidalavi Kalady, Ajeesh Elikkottil, and Rajarshi Das. 2010. Natural language question generation using syntax and keywords. In *Proceedings of QG2010: The Third Workshop on Question Generation*. questiongeneration. org, 1–10.

21. Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2015. Skip-Thought Vectors. *arXiv preprint arXiv:1506.06726* (2015).

22. Aniket Kittur, Ed H Chi, and Bongwon Suh. 2008. Crowdsourcing user studies with Mechanical Turk. In *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 453–456.

23. Aniket Kittur, Ed H Chi, and Bongwon Suh. 2009. What's in Wikipedia?: mapping topics and conflict using socially annotated category structure. In *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 1509–1512.

24. Dan Klein and Christopher D Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*. Association for Computational Linguistics, 423–430.

25. Chinmay Kulkarni, Julia Cambre, Yasmine Kotturi, Michael S Bernstein, and Scott R Klemmer. 2015. Talkabout: Making distance matter with small groups in massive classes. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*. ACM, 1116–1128.

26. Yi-Chien Lin, Li-Chun Sung, and Meng Chang Chen. 2007. An automatic multiple-choice question generation scheme for english adjective understanding. In *Workshop on Modeling, Management and Generation of Problems/Questions in eLearning, the 15th International Conference on Computers in Education (ICCE 2007)*. 137–142.

27. Sam G McFarland. 1981. Effects of question order on survey responses. *Public Opinion Quarterly* 45, 2 (1981), 208–215.

28. Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

29. George A Miller. 1995. WordNet: a lexical database for English. *Commun. ACM* 38, 11 (1995), 39–41.

30. Ruslan Mitkov, Le An Ha, and Nikiforos Karamanis. 2006. A computer-aided environment for generating multiple-choice test items. *Natural Language Engineering* 12, 02 (2006), 177–194.

31. Jack Mostow and Hyeju Jang. 2012. Generating diagnostic multiple choice comprehension cloze questions. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*. Association for Computational Linguistics, 136–146.

32. David J Nicol and Debra Macfarlane-Dick. 2006. Formative assessment and self-regulated learning: A model and seven principles of good feedback practice. *Studies in higher education* 31, 2 (2006), 199–218.

33. Andreas Papasalouros, Konstantinos Kanaris, and Konstantinos Kotis. 2008. Automatic Generation Of Multiple Choice Questions From Domain Ontologies.. In *e-Learning*. Citeseer, 427–434.

34. Lindsey E Richland, Nate Kornell, and Liche Sean Kao. 2009. The pretesting effect: Do unsuccessful retrieval attempts enhance learning? *Journal of Experimental Psychology: Applied* 15, 3 (2009), 243.

35. Henry L Roediger and Jeffrey D Karpicke. 2006a. The power of testing memory: Basic research and implications for educational practice. *Perspectives on Psychological Science* 1, 3 (2006), 181–210.

36. Henry L Roediger and Jeffrey D Karpicke. 2006b. Test-enhanced learning taking memory tests improves long-term retention. *Psychological science* 17, 3 (2006), 249–255.

37. Douglas LT Rohde. 2004. Tgrep2 user manual. (2004).

38. Vasile Rus, Brendan Wyse, Paul Piwek, Mihai Lintean, Svetlana Stoyanchev, and Cristian Moldovan. 2010. The first question generation shared task evaluation challenge. In *Proceedings of the 6th International Natural Language Generation Conference*. Association for Computational Linguistics, 251–257.

39. Caitlin Sadowski, Kathryn T Stolee, and Sebastian Elbaum. 2015. How developers search for code: a case study. In *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering*. ACM, 191–201.

40. Simon Smith, PVS Avinesh, and Adam Kilgarriff. 2010. Gap-fill tests for language learners: Corpus-driven item generation. In *Proceedings of ICON-2010: 8th International Conference on Natural Language Processing*.

41. Weiming Wang, Tianyong Hao, and Wenyin Liu. 2008. Automatic question generation for learning evaluation in medicine. In *Advances in Web Based Learning–ICWL 2007*. Springer, 242–251.

42. John H. Wolfe. 1976. Automatic Question Generation from Text - an Aid to Independent Study. In *Proceedings of the ACM SIGCSE-SIGCUE Technical Symposium on Computer Science and Education (SIGCSE '76)*. 104–112.

43. Xuchen Yao, Gosse Bouma, and Yi Zhang. 2012. Semantics-based question generation and implementation. *Dialogue & Discourse* 3, 2 (2012), 11–42.