Questimator:
Generating Knowledge Assessments for Arbitrary Topics

Qi Guo
May 4, 2016

Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Prof. Dr. Emma Brunskill
Prof. Dr. Jeffrey Bigham
Shayan Doroudi

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Robotics.

CMU-RI-TR-16-04

Copyright © 2016 Qi Guo

Keywords: Education, Knowledge Assessment, Mltiple-Choice Question

For my parents

Abstract

Formative assessments allow learners to quickly identify knowledge gaps. In traditional educational settings, expert instructors can create assessments, but in informal learning environment, 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 questions, answers and distractor options. In a study with 833 participants from Mechanical Turk, 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. Also Questimator generates questions with comparable discriminatory power as existing online quizzes. Our results suggest Questimator may be useful for assessing learning in topics for which there is not an existing quiz.

Acknowledgments

First and foremost, I would like to express my sincere gratitude to my advisor Prof. Emma Brunskill who has introduced me to rigorous research, and been really supportive through the ups and downs during the two years. I am also deeply grateful for my other research mentors, Prof. Jeffrey Bigham, Prof. Niki Kittur and Prof. Chinmay Kulkani, without whom this thesis would not have been possible. You are all great researchers. Your guidance, encouragement and knowledge have been invaluable assets for my research. It was a great pleasure to work with you all.

Further, my sincere thanks go to my amazing labmates and friends, from whom I have learned how to be a better person, not to mention the useful discussions and loads of fun. You know I am talking about you. I wish you all the best in your future endeavor.

I also gratefully acknowledge the assistance and helpful feedback of more than 800 participants from Amazon Mechanical Turk. Thanks for taking a quiz seriously.

I should probably give some credit to the nice view from our lab, which has nice chemistry with the volatile Pittsburgh weather.

My research was supported by a CMU ProSEED/Simon Initiative Seed Grants and a Google focused research award.

Contents

1 Introduction 1

2 Related Work 5

2.1 Automatic MCQ Generation 5
2.2 Ontology-based MCQ Generation 6
2.3 Crowdsourcing Question Generation 6

3 Questimator System 7

3.1 Stage 1: Related topic identification 7
3.1.1 Wikilink term frequency 8
3.1.2 Backlink Overlap 8
3.1.3 Embedded semantic similarity of topics 8
3.1.4 Combine and Rank Relatedness 9
3.2 Stage 2: Single question generation 9
3.3 Stage 2(a): Question stem generation 10
3.4 Stage 2(b): Distractor Generation 10
3.4.1 Pre-Extraction: Distractor Topic Selection 10
3.4.2 Category information 11
3.4.3 Embedded semantic similarity of topics 11
3.4.4 Post-Extraction: Distractor Phrase Ranking 11
3.5 Questimator MCQ Example 12

4 Evaluation 13

4.1 Experimental Setup 14
4.1.1 Choosing topics 14
4.1.2 Creating quizzes 14
4.1.3 Selecting questions 15
4.1.4 Participants 15
4.2 Experimental procedure 15
4.3 Objective 1: Correlation with Expert Quizzes 16
4.4 Objective 2: Question Discriminative Power 17
4.4.1 Two stage IRT model fitting 18

5 Discussion **21**
5.1 Limitations in the current Questimator system 21
5.2 Test taking strategies 21
5.3 Promising approaches that were not effective 22
5.4 What are Questimator-quizzes suited for? 23

6 Conclusion and Future Work **25**
6.1 A broader variety of questions 25
6.2 Using crowds to improve question quality 25
6.3 Learning through testing 25

Bibliography **27**

List of Figures

1.1 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.) 1

1.2 When learners enter a topic, Questimator extracts the wiki links on the page and finds 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. 3

4.1 Evaluation Experiment Interface. Crowdworkers are asked to answer mingled expert/Questimator questions one by one. 16

4.2 IRT curves of Questimator questions. Each curve corresponds to a question. Larger α, steeper curve. 19

5.1 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”). 22

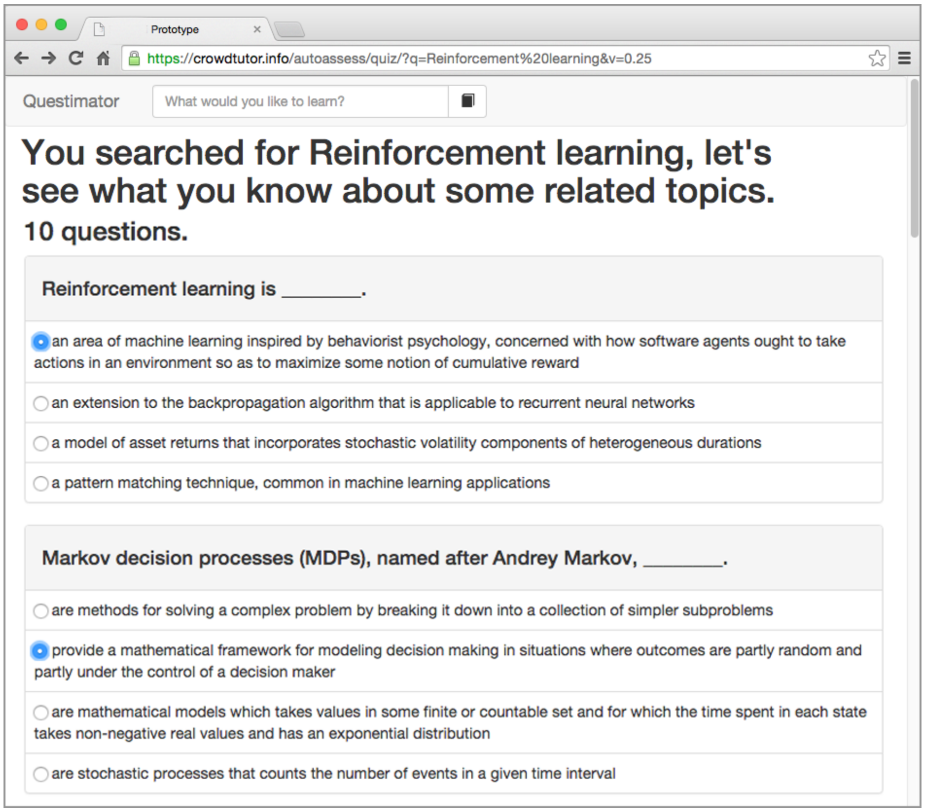
List of Tables

4.1 Expert Quiz Sources. 15
4.2 Pearson correlations of quiz scores, ordered by correlation values. 17
4.3 First column: Pearson correlations of quiz scores. Second and third columns: Mean of the questions' αs in a quiz. 19

Chapter 1

Introduction

An increasing number of learners are opportunistic and self-driven [5, 7, 34]. Online learning opportunities such as MOOCs attract diverse students with a range of prior knowledge and experience [22]. While online resources 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 [17].


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 1.1: 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.)

Assessments can help learners diagnose what they know, which may be particularly helpful in informal learning domains, so as to direct their learning in more effective ways. Knowing about their knowledge gap could help them direct their learning to first understand foundational topics before tackling more complex topics. Unfortunately informal learning domains are precisely those that often lack formal assessments. Relying on learners' self-diagnosis is tricky because they don't know what they don't know [8]. Furthermore, learners tend to overestimate the degree of their factual knowledge, leading to over-confidence [11].

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 [12]. Because creating formative assessments of factual knowledge is time-consuming, such assessments are limited to a small set of common topics of interest. Furthermore, such assessments are often missing for new or esoteric 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 (described in Section 3), which, to our knowledge, is the first system that automatically generates multiple-choice question (MCQ) quizzes for general topics from large networked corpora like Wikipedia, thus significantly extending the scope of automatically generated questions. For example, Figure 1.1 shows Questimator's automatically generated assessment for the topic of "Reinforcement learning".

A learner uses Questimator by inputting a topic they would like to learn about, *e.g.* "reinforcement learning" or "France". To generate a coherent quiz, we need questions covering different perspectives of that topic. Unfortunately, previous work (Section 2) on MCQ generation did not provide a principled approach. We propose a method (*contribution 1*) to find topics that are related to the input topic (Section 3.1) and generate questions for those related topics. We utilize the Wikipedia page-to-page linkage graph and word embedding in the process.

Generating a single multiple-choice question (Section 3.2) involves creating three pieces of output: *(i)* a question stem, *(ii)* a correct answer, and *(iii)* a set of distractors (usually, at least 3). A question stem (Section 3.3) is the part where the item to ask for is stated, for example "Reinforcement learning is _____." The examinee then chooses from correct answer and distractors. We propose a novel method (*contribution 2*) to find distractors (Section 3.4), combining Wikipedia's categorical structure (treated as a bipartite graph between the set of categorical labels and the set of articles), word embedding and sentence embedding.

In a controlled experiment with 833 participants from Mechanical Turk, we found that participants' scores on Questimator's generated quizzes are significantly and positively correlated with their scores on existing online quizzes across a variety of topics, with correlations of similar magnitude to those between existing quizzes. For certain factually-oriented topics (such as customer satisfaction), this correlation was approximately the same as score correlations between two existing online quizzes.

By strengthening related topic search and distractor generation with Wikipedia's graph structure and semantic embedding, Questimator advances question generation for general topics. Through Questimator, we demonstrate the feasibility of leveraging existing corpora of knowledge such as Wikipedia to create interactive testing materials.

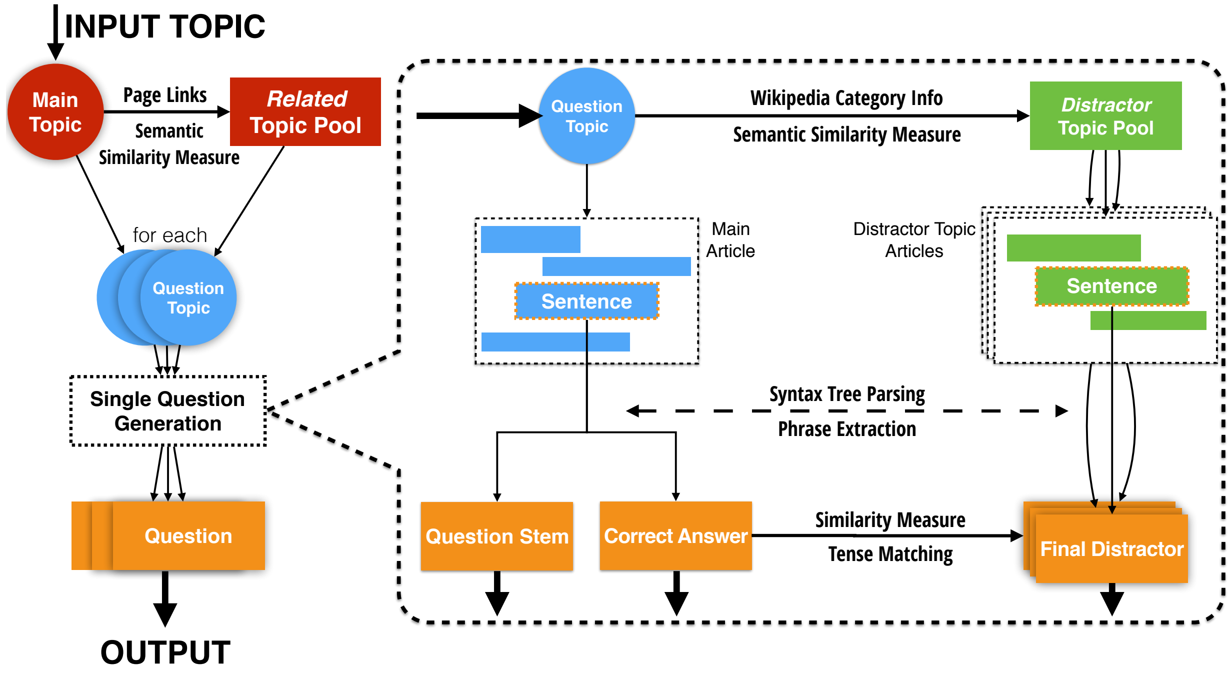


Figure 1.2: When learners enter a topic, Questimator extracts the wiki links on the page and finds related topics. A question is then generated for both the input topic and each of the related topics. To generate a question from a topic, categories from the Wikipedia page for the topic are used to generate 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.

Chapter 2

Related Work

2.1 Automatic MCQ Generation

Questimator is most closely related to 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 are derived from this article (often from a single sentence). And they mostly aimed at language learning [6, 15, 18, 23, 27, 28, 35] or a single subject/textbook [1, 3, 36].

These prior approaches to generating MCQ quizzes are limited to using single document or a fixed ontology alone to generate questions and distractors. Generating questions from a single document often scopes questions too narrowly to assess a learner's understanding of a general topic, which typically spans related documents. Related topic selection is addressed in Section 3.1.

Also, while previous approaches work well for language learning or single subject, they suffer from limited choices of distractors when directly applied to general topics. Some systems pick distractors (having same POS tag as the correct answer) within the same article/textbook, based on term frequency etc. [1, 18]. Using terms from the article as distractors can be problematic, because good distractors 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". Other systems use fixed ontologies or dictionaries, such as *WordNet*[1] [26], to find synonyms or other related words to use as distractors [15, 27].

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. This limitation also applies to other ontologies and dictionaries. In comparison, there are more than 5,129,000 English Wikipedia articles (by April 2016) for a broad range, containing long-tail topics and contemporary topics. We use the abundant Wikipedia corpora along with its socially-annotated categorical information [21] to select distractor topics. So Questimator can work across the wide range of domains. In additional it integrates semantic embeddings when generating distractors.

[1] WordNet group nouns, verbs, adjectives and adverbs into sets of cognitive synonyms (synsets). It also encodes hyperonymy (super-subordinate relation).

These approaches have therefore mostly been applied to language (vocabulary) learning and assessment [6, 15, 18, 23, 27, 35], where being able to tell similar words apart is one of the main tasks for students. A few specifically systems have also been developed for domains with domain-specific dictionary (knowledge base), like biology [1] and medicine [36].

2.2 Ontology-based MCQ Generation

Domain-specific ontologies have also been used to generate multiple-choice questions [2, 9, 30]. 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, Chaudhri et al. [9] 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 [21] 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.

2.3 Crowdsourcing Question Generation

Crowdsourcing has recently been explored as a way to generate questions. Christoforaki and Ipeirotis [10] 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. And crowdsourcing can be used for helping with searching. However this subjects to extra economical cost, human selection of topics and additional time delay. Another 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 is 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.

Chapter 3

Questimator System

Questimator is our system for automatically generating multiple choice questions from Wikipedia. To use Questimator, learners first enter a topic they would like to know about (Questimator matches 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 learner's knowledge of the provided topic (Figure 1.1 shows the first two questions generated for the topic "Reinforcement learning"). The number of questions to return is configurable (10 by default). To do this, Questimator:

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

3.1 Stage 1: Related topic identification

Related topics for questions can be drawn from prerequisites and subtopics. Given an input (main) topic X_0, to generate candidate related topics $X_i(i = 1, \ldots, n)$, Questimator leverages both the Wikipedia document for the main topic, and the larger Wikipedia structure to generate and rank related topics.

To rank a candidate related topic T, Questimator uses three measures of similarity:

- Term frequency of T in the article of X_0
- backlink overlap between T and X_0
- embedded semantic similarity between T and X_0

We will elaborate on each of the three features in the following. Questimator combines these measures to produce a single relatedness metric. And Questimator then ranks topics based on that, and chooses the top $n(n \geq 9)$ items as related topics to generate questions for the quiz.

3.1.1 Wikilink term frequency

Important concepts are likely to be repeatedly mentioned in the *main article* of X_0, and so term frequency can be an important feature for identifying important subtopics. We extract all the links to other Wikipedia topics in the main article (called *pagelinks* or *wikilinks*), and count the term frequencies ($\text{TF}_{X_0}(T)$) of the wikilinks. We take simple correferences into account, like abbreviations used in the article etc. 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 identify 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”. Related topics are also not comprehensively included in the article, for example, ”Convex Optimization” for ”SVM”.

3.1.2 Backlink Overlap

The topics that co-occur with the main topic frequently are likely to be related. Similar backlinks, i.e. articles of which topics mention the topic, may imply topic relation. We find the backlinks of the candidate topics and the main topic, and compute the cosine similarity between their “bag-of-backlinks” vectors using the following formula:

$$\text{sim}_{\text{bl}}(T, X_0) = \frac{\overrightarrow{\text{backlinks}(T)} \cdot \overrightarrow{\text{backlinks}(X_0)}}{\|\overrightarrow{\text{backlinks}(T)}\|_2 \cdot \|\overrightarrow{\text{backlinks}(X_0)}\|_2} \tag{3.1}$$

$$= \frac{|\text{backlinks}(T) \cap \text{backlinks}(X_0)|}{\|\overrightarrow{\text{backlinks}(T)}\|_2 \cdot \|\overrightarrow{\text{backlinks}(X_0)}\|_2} \tag{3.2}$$

, where $|\cdot|$ is the cardinality of the set, X_0 is the main topic, T is any candidate topic. Note here we are representing the $\overrightarrow{\text{backlinks}(T)}$ as “bag-of-backlinks” (in analogy to “bag-of-words”) vectors. So cosine similarity is the typical metric to use. Jaccard index is also reasonable if we simply consider them as two sets.

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”.

As one of the implementation details, we use the pagelink dataset dumped from Wikipedia, instead extracting the linkage information from the article database at the run time. This database contains only information about which topic links to which. It is specifically used for fast backlink queries. 1

3.1.3 Embedded semantic similarity of topics

We used a Word2Vec [25] 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 T and X_0 by their vectors' cosine similarity,

$$\text{sim}_{\text{word2vec}}(T, X_0) = \frac{\text{vec}(T) \cdot \text{vec}(X_0)}{\|\text{vec}(T)\|_2 \cdot \|\text{vec}(X_0)\|_2} \tag{3.3}$$

. If we cannot retrieve a vector for a title as an entity, we use the average of vectors for each word within the title.

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[25] model trained on Wikipedia articles with wikilinks treated as independent entities. This way, we have a direct and supposedly more accurate representation for the titles. And this will result in a more precise measurement of the semantic similarities between the topics. Due to various reasons, the Word2Vec model may not have an entity vector for every article title. For example, the dumped Wikipedia articles the Wor2Vec was trained on is out-of-date, 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.

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 semantically related, as an important optimization method for neural networks.

3.1.4 Combine and Rank Relatedness

We combine the above three quantities, to produce a single scalar used for ranking. We normalize each of the three features independently by $\tilde{f} = \frac{f - \text{mean}(f)}{\sqrt{\text{var}(f)}}$. Then they are truncated to $[-1, 1]$, so a single feature would not play a too significant role in some edge cases. We sum the three features to create a combined relatedness measure,

$$R_{X_0}(T) = \widetilde{\text{TF}}_{X_0}(T) + \widetilde{\text{sim}}_{\text{bl}}(T, X_0) + \widetilde{\text{sim}}_{\text{word2vec}}(T, X_0)$$

. These features can be assigned different weights, here we simply set them to be equally weighted. Links are finally sorted by their relatedness values, and the top n (9 by default) are picked as related topics to generate questions. For example, the related topics picked for "Reinforcement learning" are "Markov decision processes", "Software agent", "Dynamic programming", "Temporal difference learning", "Supervised learning", "Machine learning", "Optimal control theory", "Gradient descent" and "Q-learning".

3.2 Stage 2: Single question generation

Recall that a multiple choice question (MCQ) is composed of a *question stem*, a single *correct answer* and a set of *distractors*. Given a topic $X_i (i = 0, \ldots, n)$ as input, Questimator generates a single MCQ as following:

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

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

3.3 Stage 2(a): Question stem generation

Questimator generates questions directly testing on concept explanation are frequently used in expert assessments, by asking about the verbal phrases after the topic noun phrases in the main clause. Questimator generates *gap-fill questions*, as the stems, avoiding the need to transform sentences to interrogative form. To generate a question stem for a given topic, Questimator retrieves the Wikipedia article of the topic, and

- finds a set of sentences that each contain the stemmed tokens of the topic string, and process them by their order in the article,
- parses a sentence into a PCFG syntax tree, and uses TGrep [33] to match the syntax tree with certain syntactic phrase patterns,
- if matched, substitutes the matched phrase with a blank to generate the question stem, and uses the matched phrase as the correct answer.

Then Questimator processes the sentences by their order in the article, as the first few sentences where the main topic is mentioned usually contain an explanation of that topic. Note the second step is similar to Heilman and Smith [16].

Additional types of question stem can be generated by simply inserting more TGrep patterns into Questimator. As a byproduct of limiting the question stem types, we avoid the problem of choosing the proper question type.

3.4 Stage 2(b): Distractor Generation

Different from previous approaches, we propose to generate distractors utilizing the Wikipedia categorical information, and word/sentence embedding methods. To generate distractors, 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 by matching patterns on syntax tree (same as extracting the correct answer in Section 3.3),
- uses skip-thought vectors [19] to find distractor phrases most similar to the right answer phrase.

Note the finding and ranking of distractor topics is different from question topic selection (section 3.1.1).

3.4.1 Pre-Extraction: Distractor Topic Selection

Unlike searching for *related topics* as question topics (Section 3.1), the criterion of similarity for selecting distractor topics is different. For example, “Camera Matrix” is a very related topic for “Camera Calibration”, by our former criteria for relatedness. But as a distractor, examinees can easily tell a *matrix*’s definition from a calibration *process*. Luckily, semantic information in the definition phrase reveals this difference, so sentence embedding can help. In general, topics at the same level of the category hierarchy are preferred. For example, “Supervised learning” and “Unsupervised learning”, “Formula One” and “Formula Two”. So categorical structure is more informative than linkage graph for this purpose.

Questimator first finds topics sharing at least one category with the question topic to construct a candidate pool. The category information is from Wikipedia’s socially annotated (noisy) category hierarchy. Then Questimator ranks the distractor topic candidates by their Word2Vec similarity (same as in Section 3.1.3) with the question topic, and selects the top ones. For n_d ($n_d = 3$ by default) distractors, Questimator intermediately select $m \cdot n_d$ distractor topics ($m = 3$ by default) to generate the distractor phrases.

3.4.2 Category information

Questimator finds categorical information by using Wikipedia’s socially annotated category hierarchy. Although the categorical information is noisy, it can be very useful when combined with other similarity measures.

3.4.3 Embedded semantic similarity of topics

The topic semantic similarity measure here is the same as the measure for related topic explained in the former section 3.1.3.

Questimator ranks the distractor topics sharing at least one category 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.

3.4.4 Post-Extraction: Distractor Phrase Ranking

After we have mn_d distractor phrases, we aim to pick the ones most difficult to distinguish from the correct answers. Questimator applies skip-thought vectors [19] to measure the similarity between the distractors and the correct answer. Like Word2Vec generating vectors for words, skip-thought generates vectors for sentences, with semantically similar sentences having similar vectors. The model is trained on 11,038 novels in 16 genres , and skip-thought used pre-trained Word2Vec models to linearly map words and extend the vocabulary [19].

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. We match the distractors’ tenses with the correct answer. This prevents the examinees from identifying them as wrong answers by tense mismatch. Here the definition of *tense* is generalized. Instead of only indicating time for the verb, tense here includes (Tense, Person, Number).

3.5 Questimator MCQ Example

Finally distractors together with the question stem and the correct answer are delivered as a whole MCQ. The following is an example of a whole MCQ (correct answer in italic):
A recurrent neural network (RNN) is _____.

- *a class of artificial neural network where connections between units form a directed cycle*
- an artificial neural network where connections between the units do not form a directed cycle
- a parallel computing paradigm similar to neural networks, with the difference that communication is allowed between neighbouring units only
- a type of artificial neural network in which an electrically adjustable resistance material is used to emulate the function of a neural synapse

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. We maintain an updated corpus of quizzes generated from the most popular Wikipedia articles at `https://crowdtutor.info`. For many topics, such as "Wonders of the World" and "Oasis (band)", no other existing quizzes are readily available. For topics like the ongoing "Syrian Civil War", expert-generated quizzes, even if available, are likely out-of-date. Finally, for some topics, such as "Fascism" and "Constructivism", we found that Questimator generated questions similar to those on existing quizzes.

Chapter 4

Evaluation

The goal of Questimator is to generate good MCQ quizzes for knowledge evaluation for arbitrary topics.

To preliminarily evaluate Questimator questions' quality, we asked 4 TAs to inspect 258 Questimator questions (each question has one labeler), with most of questions in their areas of expertise. 78% were labeled useful for assessment. Problems with the others included multiple correct answers (36%), irrelevant question topic(22%), obvious answer (21%), all wrong answers (20%) and typos (11%). This kind of labeling has been used in previous work (Section 2). While it gives an idea how effective Questimator is at generating reasonable questions, 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 to expert-generated quizzes. Note that experimental comparisons with previous work are too difficult to run due to the lack of shared resources. We focused on two key measures:

1. the correlation between a student's performance on a set of Questimator-generated questions, and the same student's performance on a set of human expert generated questions (Section 4.3), and
2. the discriminatory power of individual Questimator-generated questions, in terms of their ability to distinguish between students' with varying topic knowledge states (Section 4.4).

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, that provides encouraging evidence that Questimator is able to capture signals that provide important insight into a student's knowledge [1]. Therefore, we focus

[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

our evaluation on comparing student performance on quizzes generated by Questimator to those generated by experts on a diverse set of topics.

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 generate many questions and then subsample the more discriminative ones with different difficult levels.

4.1 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.

4.1.1 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: customer satisfaction (marketing), earthquake (earth science), developmental psychology (psychology), cell (biology), market structure (economics), Vietnam War (history), metaphysics (philosophy), stroke (medicine), waste management (environmental science), elasticity (physics). 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 (Vietnam War and earthquake), and one for the other topics (Table 4.1).

4.1.2 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.

of how the algorithm works or how to implement it.

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

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 4.1: Expert Quiz Sources.

4.1.3 Selecting questions

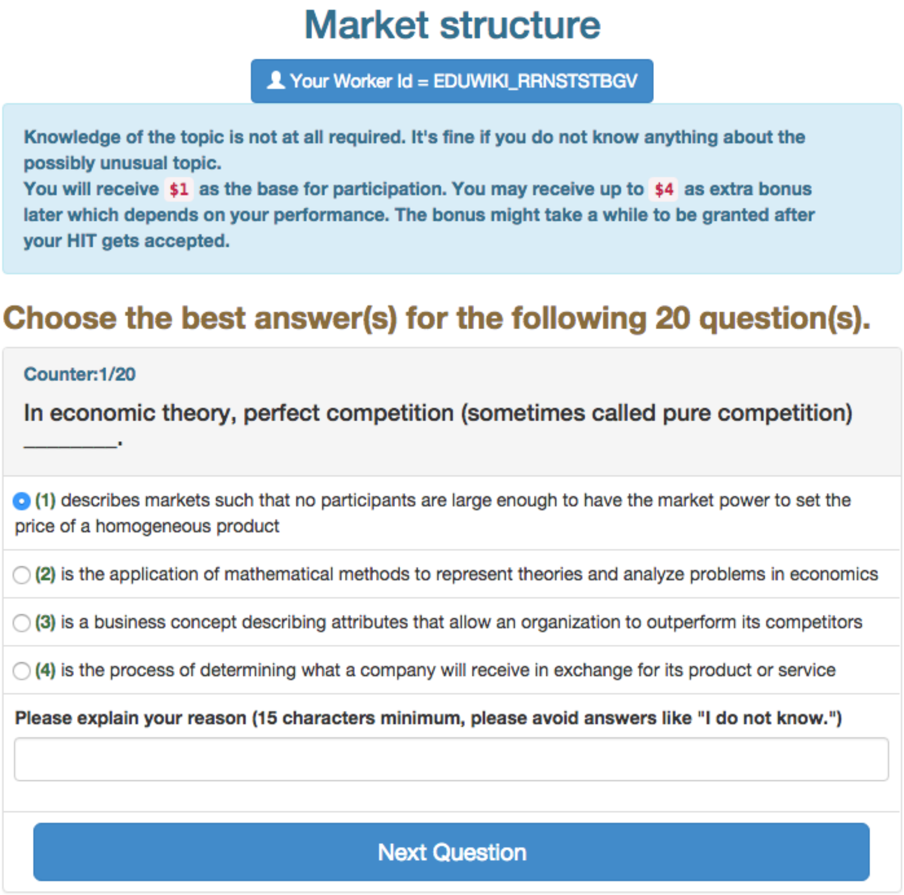
Expert-generated quizzes varied in their length from 10 to 60 questions. To eliminate testing differences across topics, we randomly sampled 10 questions from each expert quiz, after removing questions that relied on numerical calculations and True-False questions wherever possible (if removing these 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 questions on domestic issues like the Civil Rights Movement were removed before sampling. For each topic, we also generated 10 questions on Questimator.

4.1.4 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 scores.

4.2 Experimental procedure

Participants were shown one question at a time (Figure 4.1). To reduce ordering effects [24], question order was randomized across participants. To reduce response-order biases [4], the order of answer choices was also randomized. To encourage participants to put in their best effort,



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

we incentivized performance with a bonus payment of up to $4. Each question was also augmented with a textbox that asked participants to explain their reasoning (at least 15 characters), a technique that has been previously shown to encourage honest effort [20]. 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.

4.3 Objective 1: Correlation with Expert 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 4.2). 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

Quiz	Correlation
Customer satisfaction	0.465
Earthquake	0.3698
Developmental psychology	0.366
Cell (biology)	0.336
Market structure	0.282
Vietnam War	0.275
Metaphysics	0.258
Stroke	0.255
Waste management	0.216
Elasticity (physics)	0.083

Table 4.2: Pearson correlations of quiz scores, ordered by correlation values.

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 0.430 for "Earthquake" (vs 0.370 for the Questimator-expert) and 0.460 for "Vietnam War" (vs 0.275 for the Questimator-expert). Figure **??** shows the scores and correlations between Questimator-expert andå expert-expert for "Earthquake", to illustrate the variability and correlation between different groups of assessment questions.

Scores on Questimator quizzes correlate with expert quizzes to a lower but similar degree as to the two expert quizzes gathered for each category. We believe this means our quiz covers a subset of the whole topic space (also partially covered by expert quiz differently) using reasonable 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.

4.4 Objective 2: Question Discriminative Power

To analyze the discriminative power of questions, we used a very popular approach from psychometrics, Item Response Theory (IRT) [14] which is used to evaluate test items (questions) and analyze test takers. We fit a two-stage IRT model to analyze questions from a quiz under investigation (Questimator quiz) against a reference quiz (expert quiz) when they are mixed together for testing.

The IRT model we use is the unidimensional dichotomous model. For a student i with ability $\theta_i \in \mathbb{R}$, the probability of success on the jth question item is

$$P(Y_j = 1|\theta_i) = \frac{1}{1 + e^{-\alpha_j \theta_i + \beta_j}}, \tag{4.1}$$

where θ_i is the student ability, α_j and β_j are question parameters. θ_i is a scalar, implying single knowledge ability for the related term affects students' performance on the quiz. Naturally α_j is also unidimensional.α_j is the question discriminative parameter, and β_j is question difficulty.

We selected this simple version of an IRT model to alleviate over-fitting. It can also be easily visualized and interpreted. For the IRT curve of an item (in our case, as question), α specifies the curve steepness (larger α, steeper curve) and β shifts the curve horizontally (larger β, more to the right). We want a steeper curve (in the extreme case, a step function), so there is less randomness in the score given a student's ability. More specifically, we considered a Bayesian IRT model with a prior of

$$\theta_i \sim N(0,1) \tag{4.2}$$

$$\alpha \sim N(\mu_\alpha = 1, \delta_\alpha^2) \tag{4.3}$$

$$\beta \sim N(\mu_\beta = 0, \delta_\beta^2) \tag{4.4}$$

. μ_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 [14]. We sample for 2000000 iterations, and burn the first 1990000 iterations.

4.4.1 Two stage IRT model fitting

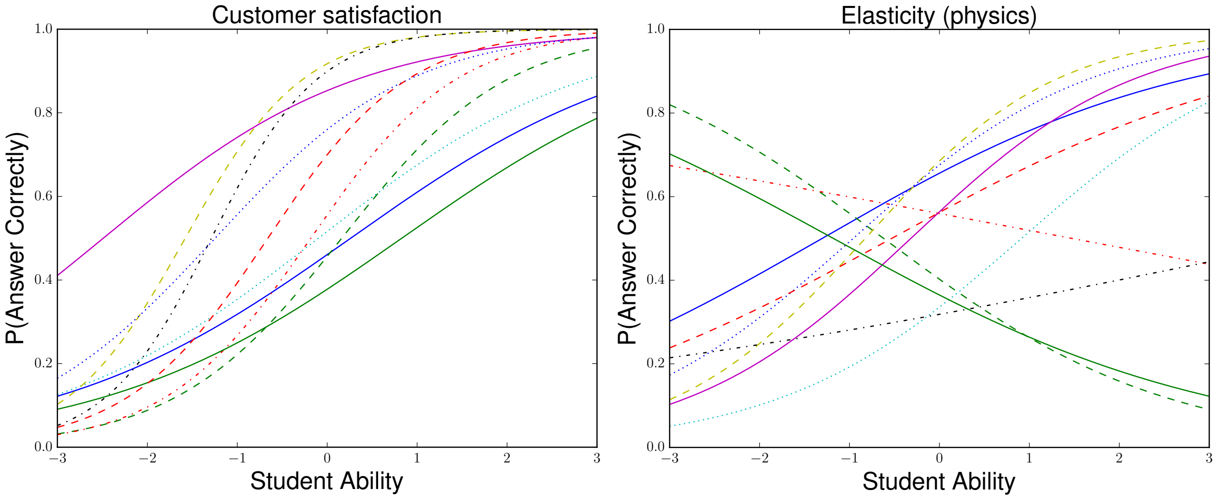
We have two sets of questions (from experts and from Questimator) and if we estimated the IRT parameters independently, it would be less clear how to compare them. Instead, we fit two IRT models for each topic (*i.e.,* each quiz mixture). We first fit one IRT model with the expert questions. We then treat the estimated student's ability $\hat{\theta}_{i,exp}$s as the true underlying student abilities of the tested topic. We fix $\theta_i = \hat{\theta}_{i,exp}$s and fit another IRT model with Questimator questions, estimating only α and β for the Questimator questions (Figure 4.2).

For "Customer satisfaction" (Pearson correlation across overall expert-Questimator question sets: 0.465), 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: 0.083), 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. With this method, we have a reliable evaluation of the quality of each question. Further investigation will help us to improve the weak aspects of the system, and to automatically eliminate problematic questions.

Table 4.3 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.

Quiz	Correlation	Expert Quiz $\bar{\alpha}$	Questimator Quiz $\bar{\alpha}$
Cell (biology)	0.336	0.834	0.549
Customer satisfaction	0.465	0.428	0.992
Development psychology	0.366	0.992	0.464
Earthquake	0.370	0.562	0.643
Elasticity (physics)	0.083	0.465	0.267
Market structure	0.282	0.439	0.616
Metaphysics	0.259	0.645	0.450
Stroke	0.255	0.337	0.638
Vietnam War	0.275	0.838	0.422
Waste management	0.216	0.652	0.548
Average	0.291	0.619	0.559

Table 4.3: First column: Pearson correlations of quiz scores. Second and third columns: Mean of the questions' αs in a quiz.



(a) Customer satisfaction

(b) Elasticity (physics)

Figure 4.2: IRT curves of Questimator questions. Each curve corresponds to a question. Larger α, steeper curve.

Chapter 5

Discussion

5.1 Limitations in the current Questimator system

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

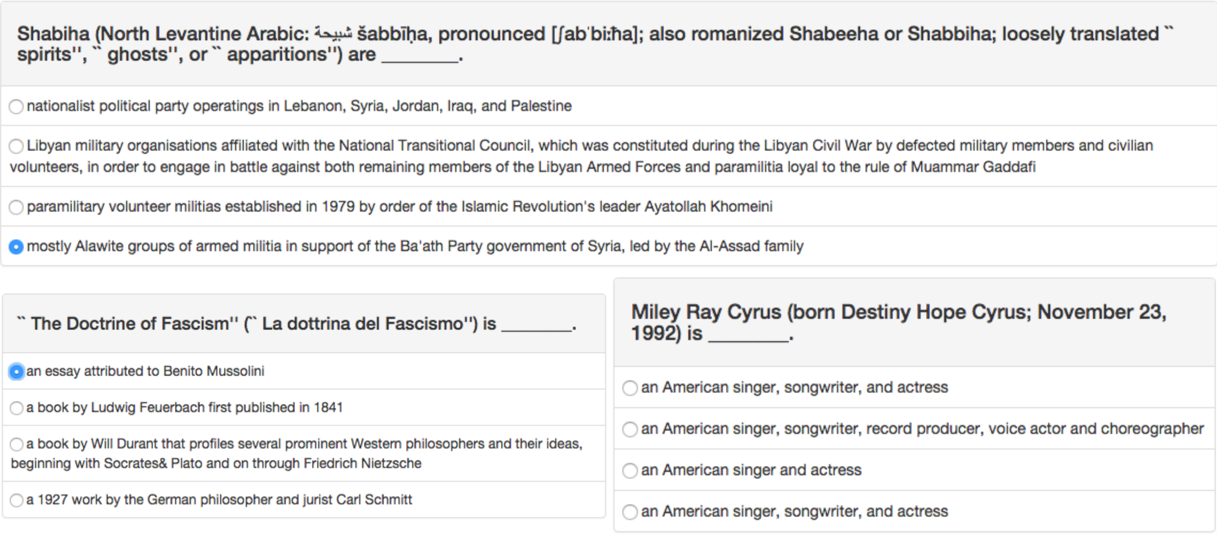
Because Questimator uses the target article's text as the starting point for generating questions, we find its ability to generate good questions depends critically on the quality of the article text itself. First, when when the target article was too short (*e.g.*, a Wikipedia "stub"), Questimator was 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 5.1, 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).

5.2 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. At



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 5.1: 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").

the end of the quiz, workers are asked to provide optional feedback: "Is anything wrong with the questions? How can we improve the question quality? How can we improve this HIT?" The main types of the feedback are: *(1)* explanation why he/she might did well or bad, *(2)* enjoying it, *(3)* finding this difficult, *(4)* questions are all good, *(5)* spotting errors in one question and *(6)* discussion of the tested topic.

5.3 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 [13], 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.

5.4 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 [29].

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.

Chapter 6

Conclusion 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, and that we can generate quizzes for many topics for which no quiz currently exists. Future work will explore *(i)* improving the quality and the breadth of automatically created assessments, *(ii)* integrating it into systems for learning through testing, and *(iii)* utilizing additional corpora.

6.1 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.

6.2 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.

6.3 Learning through testing

Testing students on knowledge they are about to gain (pre-testing) can enhance learning [31, 32]. 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.

Bibliography

[1] Manish Agarwal and Prashanth Mannem. Automatic gap-fill question generation from text books. In *Proceedings of the 6th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 56–64. ACL, 2011. 2.1

[2] Maha Al-Yahya. Ontoque: a question generation engine for educational assesment based on domain ontologies. In *Advanced Learning Technologies (ICALT), 2011 11th IEEE International Conference on*, pages 393–395. IEEE, 2011. 2.2

[3] Arjun Singh Bhatia, Manas Kirti, and Sujan Kumar Saha. Automatic generation of multiple choice questions using wikipedia. In *Pattern Recognition and Machine Intelligence*, pages 733–738. Springer, 2013. 2.1

[4] Niels J Blunch. Position bias in multiple-choice questions. *Journal of Marketing Research*, pages 216–220, 1984. 4.2

[5] Joel Brandt, Philip J Guo, Joel Lewenstein, Mira Dontcheva, and Scott R Klemmer. Two studies of opportunistic programming: interleaving web foraging, learning, and writing code. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1589–1598. ACM, 2009. 1

[6] Jonathan C Brown, Gwen A Frishkoff, and Maxine Eskenazi. Automatic question generation for vocabulary assessment. In *Proceedings of the conference on HLT and EMNLP*, pages 819–826. ACL, 2005. 2.1

[7] Carrie J Cai, Philip J Guo, James R Glass, and Robert C Miller. Wait-learning: Leveraging wait time for second language education. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3701–3710. ACM, 2015. 1

[8] Deanna Caputo and David Dunning. What you don't know: The role played by errors of omission in imperfect self-assessments. *Journal of Experimental Social Psychology*, 41(5): 488–505, 2005. 1

[9] Vinay K Chaudhri, Peter E Clark, Adam Overholtzer, and Aaron Spaulding. Question generation from a knowledge base. In *Knowledge Engineering and Knowledge Management*, pages 54–65. Springer, 2014. 2.2

[10] Maria Christoforaki and Panagiotis G Ipeirotis. A system for scalable and reliable technical-skill testing in online labor markets. *Computer Networks*, 2015. 2.3

[11] Heidi S Chumley-Jones, Alison Dobbie, and Cynthia L Alford. Web-based learning: Sound educational method or hype? a review of the evaluation literature. *Academic medicine*, 77

(10):S86–S93, 2002. 1

[12] Jannette Elwood and Val Klenowski. Creating communities of shared practice: the challenges of assessment use in learning and teaching. *Assessment & Evaluation in Higher Education*, 27(3):243–256, 2002. 1

[13] Günes Erkan and Dragomir R Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, pages 457–479, 2004. 5.3

[14] Jean-Paul Fox. *Bayesian item response modeling: Theory and applications*. Springer Science & Business Media, 2010. 4.4, 4.4

[15] Donna Marie Gates, Greg Aist, Jack Mostow, Margaret McKeown, and Juliet Bey. How to generate cloze questions from definitions: A syntactic approach. In *2011 AAAI Fall Symposium Series*, 2011. 2.1

[16] Michael Heilman and Noah A Smith. Good question! statistical ranking for question generation. In *NAACL/HLT 2010*. ACL, 2010. 3.3

[17] Elaine K Horwitz, Michael B Horwitz, and Joann Cope. Foreign language classroom anxiety. *The modern language journal*, 70(2):125–132, 1986. 1

[18] Ayako Hoshino and Hiroshi Nakagawa. A real-time multiple-choice question generation for language testing: A preliminary study. In *EdAppsNLP*, pages 17–20, Stroudsburg, PA, USA, 2005. ACL. 2.1

[19] Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Skip-thought vectors. *arXiv preprint arXiv:1506.06726*, 2015. 3.4, 3.4.4

[20] Aniket Kittur, Ed H Chi, and Bongwon Suh. Crowdsourcing user studies with mechanical turk. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 453–456. ACM, 2008. 4.2

[21] Aniket Kittur, Ed H Chi, and Bongwon Suh. 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*, pages 1509–1512. ACM, 2009. 2.1, 2.2

[22] Chinmay Kulkarni, Julia Cambre, Yasmine Kotturi, Michael S Bernstein, and Scott R Klemmer. Talkabout: Making distance matter with small groups in massive classes. In *Proceedings of the 18th ACM CSCW*, pages 1116–1128. ACM, 2015. 1

[23] Yi-Chien Lin, Li-Chun Sung, and Meng Chang Chen. 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 ICCE*, pages 137–142, 2007. 2.1

[24] Sam G McFarland. Effects of question order on survey responses. *Public Opinion Quarterly*, 45(2):208–215, 1981. 4.2

[25] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013. 3.1.3, 3.1.3

[26] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38 (11):39–41, 1995. 2.1

[27] Ruslan Mitkov, Le An Ha, and Nikiforos Karamanis. A computer-aided environment for generating multiple-choice test items. *Natural Language Engineering*, 12(02):177–194, 2006. 2.1

[28] Jack Mostow and Hyeju Jang. Generating diagnostic multiple choice comprehension cloze questions. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 136–146. ACL, 2012. 2.1

[29] David J Nicol and Debra Macfarlane-Dick. Formative assessment and self-regulated learning: A model and seven principles of good feedback practice. *Studies in higher education*, 31(2):199–218, 2006. 5.4

[30] Andreas Papasalouros, Konstantinos Kanaris, and Konstantinos Kotis. Automatic generation of multiple choice questions from domain ontologies. In *e-Learning*, pages 427–434. Citeseer, 2008. 2.2

[31] Lindsey E Richland, Nate Kornell, and Liche Sean Kao. The pretesting effect: Do unsuccessful retrieval attempts enhance learning? *Journal of Experimental Psychology: Applied*, 15(3):243, 2009. 6.3

[32] Henry L Roediger and Jeffrey D Karpicke. Test-enhanced learning taking memory tests improves long-term retention. *Psychological science*, 17(3):249–255, 2006. 6.3

[33] Douglas LT Rohde. Tgrep2 user manual, 2004. 3.3

[34] Caitlin Sadowski, Kathryn T Stolee, and Sebastian Elbaum. How developers search for code: a case study. In *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering*, pages 191–201. ACM, 2015. 1

[35] Simon Smith, PVS Avinesh, and Adam Kilgarriff. Gap-fill tests for language learners: Corpus-driven item generation. In *Proceedings of ICON*, 2010. 2.1

[36] Weiming Wang, Tianyong Hao, and Wenyin Liu. Automatic question generation for learning evaluation in medicine. In *Advances in Web Based Learning–ICWL 2007*, pages 242–251. Springer, 2008. 2.1