Key Phrase Extraction for Generating Educational Question-Answer Pairs

Angelica Willis
Stanford University
Stanford, CA, USA
arwillis@stanford.edu

Glenn Davis
Stanford University
Stanford, CA, USA
gmdavis@stanford.edu

Sherry Ruan
Stanford University
Stanford, CA, USA
ssruan@stanford.edu

Lakshmi Manoharan
Stanford University
Stanford, CA, USA
mlakshmi@stanford.edu

James Landay
Stanford University
Stanford, CA, USA
landay@stanford.edu

Emma Brunskill
Stanford University
Stanford, CA, USA
ebrun@stanford.edu

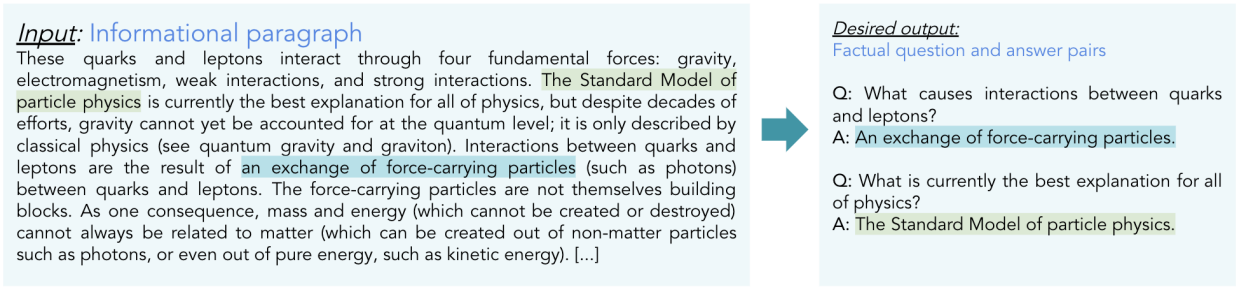


Figure 1: With Key Phrase Extraction, any informational passage can be converted into a quiz-like learning module.

ABSTRACT

Automatic question generation is a promising tool for developing the learning systems of the future. Research in this area has mostly relied on having answers (key phrases) identified beforehand and given as a feature, which is not practical for real-world, scalable applications of question generation. We describe and implement an end-to-end neural question generation system that generates question and answer pairs given a context paragraph only. We accomplish this by first generating answer candidates (key phrases) from the paragraph context, and then generating questions using the key phrases. We evaluate our method of key phrase extraction by comparing our output over the same paragraphs with question-answer pairs generated by crowdworkers and by educational experts. Results demonstrate that our system is able to generate educationally meaningful question and answer pairs with only context paragraphs as input, significantly increasing the potential scalability of automatic question generation.

LAS'19, June 24–25, 2019, Chicago, IL, USA

ISBN 978-1-4503-2138-9…$0.00

DOI: 10.1145/3330430.3333636

ACM Classification Keywords

J.1 Computer Applications: Education; I.2.7 Artificial Intelligence: Natural Language Processing; H.3.3 Information Storage and Retrieval: Information Search and Retrieval

Author Keywords

Automatic answer extraction; Educational content generation; Recurrent neural networks; Educational question generation

INTRODUCTION

For educators, questioning students to assess and reinforce learning is a key component of effective teaching. Questioning not only confirms the acquisition of knowledge, it also aids critical thinking, retention, and engagement. Similarly, technology-based educational systems must be able to produce legible, pedagogically-salient questions to deliver meaningful learning experiences. Indeed, prior work has proposed automatic question generation for a variety of educational use cases, such as academic writing support [7], reading comprehension assessment [11], and educational chatbots [10]. The typical goal of these projects is to take a passage of text and generate a grammatical and sensible question that serves some pedagogical purpose; however, these systems typically rely on simple rules for selecting the pivot word or phrase that will be used as the answer for the generated question.

As such, the limitations of these systems make it challenging to assess the level of understanding of the student. Although

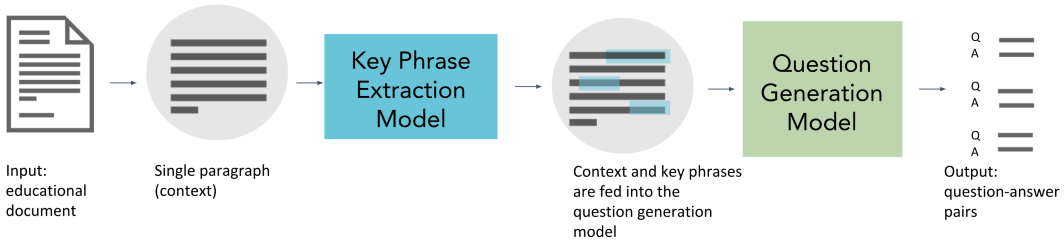


Figure 2: Overview of a proposed system that could take nearly any education document (textbook, Wikipedia article, children's story) and create pedagogical-quality, domain specific question and answer pairs

they may be able to generate novel questions, their scope is limited by the rule-based selection methods for the content of the question. To our knowledge, no previous system has been evaluated for the task of assessing the text passage and identifying a relevant and pedagogically valuable question, without the answer to that question already provided. Key phrase extraction is a vital step to allow automatic question generation to scale beyond datasets with predefined answers to real-world education applications.

In this paper, we describe an end-to-end question generation process that takes as input "context" paragraphs from the Stanford Question Answering Dataset (SQuAD) [14], initially sourced from Wikipedia, and outputs factual question and answer pairs based on the contexts[1]. In order to accomplish this task, we first generate answer candidates from the contexts, allowing us to generate questions using the answer candidates on any type of informational text. We use this information to produce pedagogically valuable questions that use key phrases from the context as answers.

We show that a generative model, even one trained only on extractive answers from SQuAD, can generalize better to key phrases generated by educational experts than traditionally used word level binary key phrase predictors. We are the first system of its kind to be assessed by domain experts (classroom teachers) to evaluate the pedagogical values of question-answers pairs generated. Most previous works have only evaluated the coherency, fluency or grammatical correctness.

RELATED WORK

We first present general question generation which has been extensively studied in the Natural Language Generation community. Then we discuss how these techniques have been applied to education to generate educational questions at scale, along with current limitations in educational question generation. Lastly, we discuss various two-stage generation models that are related to our model.

General Question Generation

We use *general question generation* to refer to generating natural and fluent general questions from a given paragraph. These generated questions can be particularly helpful for constructing labeled datasets for machine reading comprehension and machine question answering research. Therefore, naturalness and the level of answer difficulty are some key evaluation criteria [2].

Traditionally, researchers leveraged deep linguistic knowledge and well-designed rules to construct question templates and generate questions [15]. More recently, deep learning based models have been developed to generate a large number of questions without hand-crafting too many rules. Serban et al. [16] used a neural network on a knowledge base to generate simple factoid questions. Du et al. [2] were the first to use a deep sequence-to-sequence learning model to generate questions. They used the sentences from SQuAD [14] containing answers as input for their neural question generating model.

Question Generation in Education

Both rule-based and deep learning based approaches have been applied to *educational question generation*.

Mitkov and Ha [9] used well-designed rules and language resources to generate multiple-choice test questions and distractors. Liu et al. [7] scanned through a student's essay to find academic citations around which probing questions are built. Mostow and Jang [11] removed the last words of passages to generate multiple-choice fill-in-the-blank "cloze" questions.

Deep learning models were also adopted to generate educational questions. Wang et al. [18] used SQuAD to build a recurrent neural network-based automatic question generator, QG-Net. They used the answers from the human-generated questions in SQuAD to build new questions with the same answers. Since deep learning models require less domain knowledge to construct rules or templates, they have greater potential to generate educational assessment questions at scale.

However, limitations exist in current deep learning models for educational question generation. Although researchers used automatic metrics such as BLEU [12], METEOR [5], and ROUGE [6], as well as human evaluators [17, 18], to assess the quality of generated educational questions, their main evaluation focus has still been on the coherency and fluency of the questions generated [18]. Few works have explored the use of educational experts to evaluate these generated questions from a pedagogical perspective.

Some previous work has attempted to conduct deeper evaluations by assessing students' performance on these questions. For example, Guo et al. [3] not only extracted and generated

[1] https://rajpurkar.github.io/SQuAD-explorer/

multiple-choice assessment questions from Wikipedia, but they also ran a study with 833 Mechanical Turkers to show the close correlation between these people's scores on generated quizzes and their scores on actual online quizzes. In our work, we instead recruited domain experts (classroom teachers) to more rigorously verify the actual pedagogical value of the content generated.

Key Phrase Extraction for Question Generation

Our work is built upon a family of two-stage generation models that first extract key phrases then generate questions based upon extracted key phrases.

Key phrase extraction (KPE) alone is an interesting research question. Meng et al. [8] proposed an encoder-decoder generative model for key phrase prediction. Their model can generate key phrases based on the semantic meaning of the text, even though the key phrases are not in the text. However, without the subsequent question generation phase, the purpose of extracting these key phrases is usually for information retrieval, text summarization, and opinion mining. Yang et al. [19] used linguistic tags and rules to extract answers from unlabeled text, and then generated questions based on extracted answers. However, their generated questions were used to improve their question generation models instead of any educational purposes.

Though in QG-Net [18] both context paragraphs and answer phrases need to be provided at the same time for the model to generate questions, Subramania et al. [17] proposed a two-stage neural model to generate questions from documents directly. They first used a neural key-phrase extractor to assign to each word sequence a probability of being a key answer. They then used a sequence-to-sequence question-generation model to generate questions conditioned on the predicted answers. They demonstrated the potential for generating fluent and answerable questions in this way, which is most related to what we first explore with a binary word classifier (KPE-Class); we then further build on that foundation with a non-extractive, generative language model (KPE-Gen).

METHODS & TECHNIQUES

In this section, we first present the data pre-processing techniques we used, and then two models for answer generation in detail: a binary classifier and a more sophisticated encoder-decoder neural network. Last, we present an end-to-end question-answer pair generation pipeline.

Answer generation involves picking candidate terms that could answer an *interesting and pedagogically meaningful* question about the passage. Although there might exist several definitions to what one might consider *interesting*, we focus our scope on those that we believe are most likely to bear relevance to knowledge acquisition, directly extracted from SQuAD context passages.

We explore two approaches to key phrase extraction: a conventional **classifier-based** approach as well as a novel **generative language model based** approach. Eventually, we see the direction of educational KPE research taking a more generative path, as the task becomes less focused on directly extracting facts from the context, and more focused on generating deeper reasoning questions.

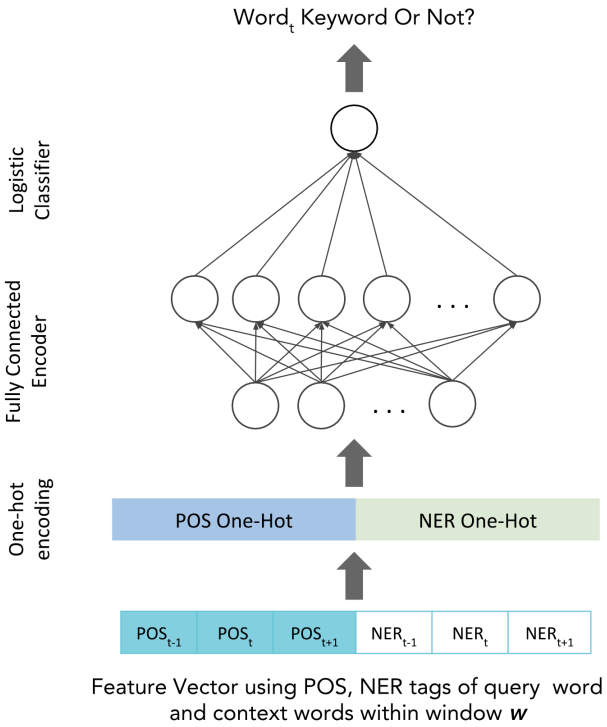


Figure 3: Architecture for second Answer Generation Model (binary classifier)

The extracted answers are then used as *answer inputs* to generate questions associated with them using a pre-trained question generation model. The entire question-answer pair generation pipeline is illustrated in Figure 2.

Pre-processing Techniques

We present two data pre-processing techniques we used: part of speech tagging and named entity recognition.

Part of speech: We used the detailed part-of-speech (POS) tagger in spaCy[2] to find all POS tags in the context passages and answer sets. For each POS tag, we divided the number of times it appears in the answer set by the number of times it appears in the context passages. This proportion indicates which POS tags are most likely to be in the answer set, and thus which POS tags are associated with key phrases in the context.

Named entity recognition: We also used the named entity recognition (NER) tagger in spaCy to find all NER tags in the context passages and answer sets. We followed the same procedure as with the previous section to determine the most important NER tags.

[2] https://spacy.io/

Binary Classifiers

We first present a simple *classifier-based* approach, called **KPE-Class**, for answer generation. KPE-Class treats the process as a series of binary classification tasks, trained to predict the probability a word is an answer from the SQuAD dataset, and thus which words from the context to extract as the answers. Specifically, for each word in a given context passage, the network outputs the probability that the given word is a keyword. We then concatenate (offline) contiguous sequences of words that are classified as potential keywords, to generate all key phrases associated with the context passage.

Feature Vector: Let the context word at the ith position be c_i. Let POS_x be defined as the *Part-Of-Speech* tag for context word at position x, and NER_x be defined as the *Named Entity Recognition* tag for context word at position x. Note that we consider the 46 POS tags and 8 NER tags as given by the *NLTK* library trained using the UPenn Corpus. We represent each of the POS/NER categories as integers, by maintaining a consistent mapping of these tags to integers. Our feature vector $\mathbf{c_i}$ is then constructed as follows: Concatenate POS_{i-w}, POS_{i-w+1},...,POS_i, POS_{i+1},...POS_{i+w}, NER_{i-w}, NER_{i-w+1},...,NER_i, NER_{i+1},...NER_{i+w}, where w is the window of surrounding context words considered. This yields a feature vector $\mathbf{c_i} \in \text{R}^{2w+1}$. We experimented with window sizes {1,2} and choose 2 empirically in favor of higher F1 score on the validation set.

Model Construction: The feature vector obtained in the previous step is then encoded as a one-hot vector, in order to use the NER/POS categorical variables in our deep neural network. This encoded representation is then fed into a fully connected layer with ReLU activation. The final layer consists of a single unit with logistic activation. This yields the probability that the given context word c_i is a keyword.

Encoder-Decoder Neural Network

We then present a novel *language model based* encoder-decoder network denoted as **KPE-Gen** for identifying key terms, leveraging OpenNMT[4]. The context tokens, represented using GloVe [13] (an unsupervised learning method for obtaining vector representations of words) embeddings, are encoded using a 2 layer bidirectional LSTM network, which is capable of preserving information from both past and future points of the sequence (see Figure 5). The architecture is illustrated in Figure 4.

We use this information to inform our model about the desired length and number of "answers" (key phrases) to generate, as well as which word-level features (part-of-speech, named entity recognition) to consider.

Encoding: Let the context passage, represented using GloVe embeddings, be $\mathbf{c_i} \in \text{R}^{n \times d}$, where n is the number of tokens in the context and d is the dimension of the GloVe embedding. Since the number of tokens vary with each context, we pad or truncate the context passages as necessary to meet $n = context\_len$, where $context\_len$ is a hyper-parameter. We further append the POS (part-of-speech) and NER (Named Entity Recognition) features to the embedded context, yielding $\mathbf{c_i} \in \text{R}^{n \times (d+2)}$, as there is 1 POS feature and 1 NER feature as-

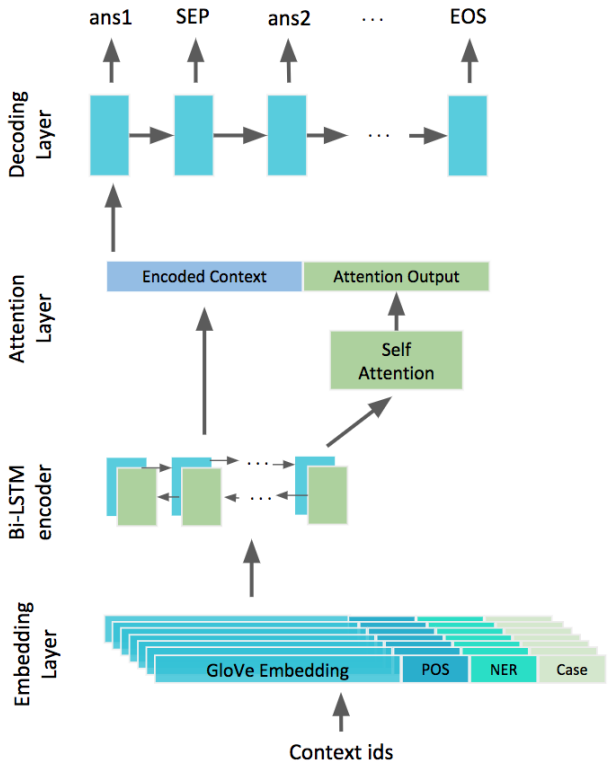


Figure 4: Our architecture for Key Phrase Extraction (language model)

sociated with each token in the context. We encode the context information thus obtained using a bidirectional LSTM decoder with h hidden states. We represent the encoded context as $\mathbf{c_e} \in \text{R}^{n \times 2h}$.

Self-attention: We then apply basic dot-product attention, with each embedded token in the context to attend to every token in the context.
Let $\mathbf{e^i}$ be the attention distribution defined as below.

$$\mathbf{e^i} = [\mathbf{c_i}^T \mathbf{c_1}, \mathbf{c_i}^{\mathbf{c_2}}, ..., \mathbf{c_i}^{\mathbf{c_n}}] \in \text{R}^n$$

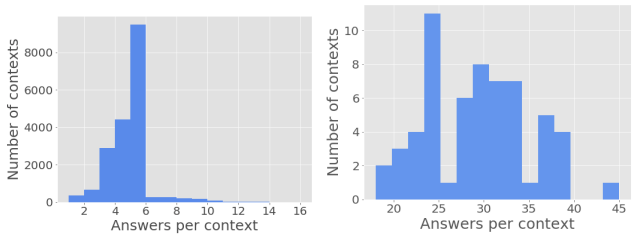
$$\alpha_i = \mathbf{softmax}(\mathbf{e^i}) \in \text{R}^n$$

Then, we can obtain the attention output $\mathbf{a_i}$ associated with each context as follows:

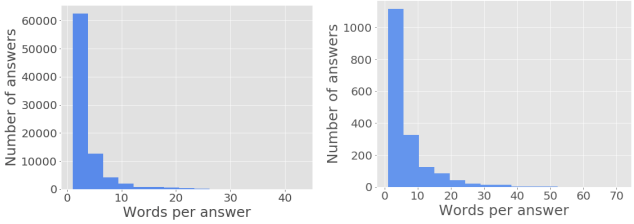
$$\mathbf{a_i} = \sum_{j=1}^{n} \alpha_j^i \mathbf{c_j} \in \text{R}^{2h}$$

We then created a combined representation of the encoded context and attention output as $\mathbf{b_i} = [\mathbf{c_i}; \mathbf{a_i}]$, where the semicolon indicates $\mathbf{a_i}$ being appended to $\mathbf{c_i}$. See Figure 8 for a visualization of the attention distribution for an example output.

The LTSM-based decoder (see Figure 4) generates all strings separated by a separated SEP, and makes the dynamic decision of when to stop generating more key phrases by producing an EOS tag.

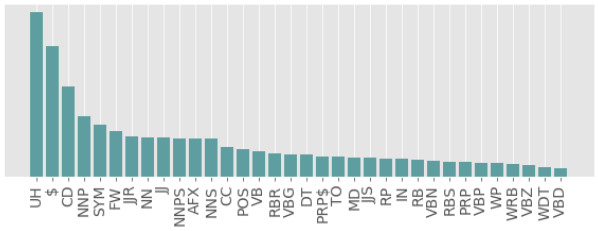


(a) SQuAD (b) Expert

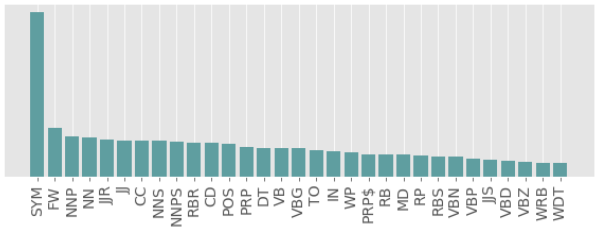


(c) SQuAD (d) Expert

Figure 5: Descriptive characteristics of SQuAD and education expert dataset. Top row: Number of answers per context. Bottom row: Number of words per answer.



(a) SQuAD



(b) Expert

Figure 6: Ranking of part-of-speech tags by over-representation in answers as compared to chance

Training: KPE-Gen was trained on the context + answer pairs of the SQuAD training set using pretrained, 300-dimensional word embeddings for 20 epochs.

DATASET

In this section, we describe the dataset we used for training and evaluation, interesting findings on this dataset, and the collection and analysis of education expert annotated data.

The Stanford Question Answering Dataset

The Stanford Question Answering Dataset (SQuAD) [14] consists of 536 articles ("context" paragraphs) extracted from Wikipedia, with 107,785 question-answer pairs generated by human crowdworkers. Figures 5a and 5c show some basic characteristics of the SQuAD training set, consisting of 442 articles. As can be seen, the average number of answers provided per context passage peaks around 5-6, and most answers are one word in length. Given the restriction that all answers are unbroken sequences of words taken from the context paragraph, it is not surprising that most of the questions and answers in SQuAD are simple fact-based questions such as *"Media production and media reception are examples of what type of context?"* with the answer *"ethnographic"*.

As discussed in the Methods & Techniques section, we preprocessed the data using part-of-speech (POS) tagging and named entity recognition (NER). For each POS and NER tag, we divided the number of times it appears in the answer set by the number of times it appears in the context passages to find the tags that are more likely to appear in answers than would be predicted by chance.

Figure 6a shows the most important POS tags in the SQuAD training set by this metric, filtering out tags that occur less than 100 times and tags associated with punctuation marks. We find that the five most important POS tags are *UH (interjection)*, *$ (currency)*, *CD (cardinal number)*, *NNP (singular proper noun)*, and *SYM (symbol)*. The five least important POS tags are *VBD (past tense verb)*, *WDT (wh-determiner)*, *VBZ (3rd-person singular present tense verb)*, *WRB (wh-adverb)*, and *WP (wh-pronoun, personal)*.

Given our earlier observation that most question-answer pairs are fact-based, the importance of currency, cardinal numbers (often representing years and/or dates), and proper nouns is not surprising. *UH (interjection)* is unexpectedly marked as the most important POS tag, but it only appears 262 times in total in the context paragraphs and 160 times in the answer set; compare with *NNP (singular proper noun)*, which appears 282,960 times in the context paragraphs and 63,995 times in the answer set. A larger sample size may be needed to determine whether interjections are indeed frequently represented in answers.

Figure 7a shows the most important NER tags in SQuAD by the same metric. For SQuAD, we find that the five most important NER tags are *MONEY*, *CARDINAL (unclassified numbers)*, *PERCENT*, *DATE*, and *PERSON*. The five least important NER tags are *PRODUCT*, *WORK_OF_ART (books, songs, etc.)*, *FAC (facilities; e.g., buildings, airports)*, *LOC (locations)*, and *LAW (named documents made into laws)*.

As with POS, tags indicating money and cardinal numbers are again marked as important, and the other important tags similarly fit well with fact-based question-answer pairs. However, the low importance of location tags is surprising, and further research into the types of questions and answers generated by crowdworkers may reveal some insight into why this is the case.

Education Expert Data

We generated an oracle dataset by asking 51 human domain experts (current or former classroom teachers) to generate reading comprehension questions of educational value based on given text passages. We first extracted 60 random context paragraphs from SQuAD, then using a Qualtrics[3] survey, we displayed 20 of these passages to each human expert in a randomized order. For each text passage, participants imagined that they were teaching a class and were assigning the text passage to be read. The participants were asked to identify the most important information in the passage, and generate 2-5 questions to test whether a student had correctly read and understood that important information in the passage.

We thus replicated the data collection procedure used to generate the answers in SQuAD [14]; however, we used domain experts to produce an answer set that represents expert selection of key phrases. Characteristics of this expert answer set and how they compare to the SQuAD answer set can be seen in Figures 5b, 5d, 6b, and 7b. We use this new answer set to validate the original SQuAD answer sets and to gauge our model's effectiveness.

In order to assess the level of agreement between the domain experts, we computed the *Jaccard index* for every pair of expert answer sets for each passage using the following equation:

$$J(A,B) = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}.$$

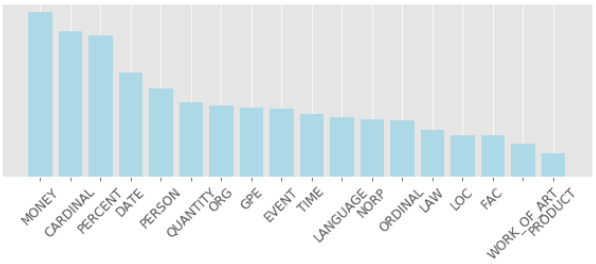
For the purpose of computing the set intersections for the Jaccard index, we consider answers with a similarity ratio $2M/T \geq 0.7$ as the same answer (and thus included in the intersection $A \cap B$), where T is the total number of elements in both answers and M is the number of matched elements. As an example, the answers "the Indian Ocean" and "Indian Ocean" have a similarity ratio of $2 \times 11/25 = 0.88$ and are considered as the same answer.

Averaging the Jaccard index for all pairwise comparisons of the 51 experts over all 60 passages, we observed an overall internal Jaccard index for the expert dataset of 0.1275. We also calculated the mean Jaccard index for all pairwise comparisons between expert answers and the SQuAD answers, and found it to be 0.1260, suggesting that the experts produce answer sets that are not much more similar to each other (i.e., internally consistent) than they are to the SQuAD answer sets. However, we see that when we aggregate all expert answers and select the top K most popular phrases, where K is the rounded average number of answers per context in SQuAD (see Key Phrase Adjudication) for our final answer set for each context, the Jaccard index increases to 0.2596, showing that consensus phrases lean closer to the SQuAD set.

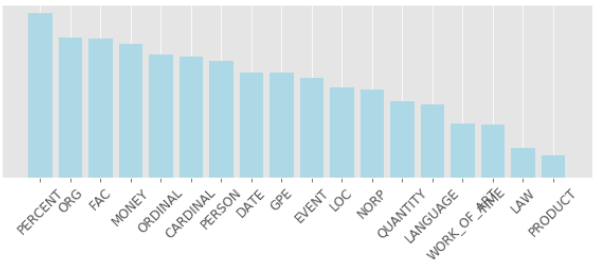
RESULTS

We evaluated our models on SQuAD ground truth data, as well as data we collected from our educational expert study, from which we generated a small evaluation set.

[3] https://www.qualtrics.com/



(a) SQuAD



(b) Expert

Figure 7: Ranking of named entity tags by over-representation in answers as compared to chance

Key Phrase Adjudication

Participants in the educational expert data collection study generated a combined total of 2,214 key phrases, for an average of 37 phrases for each of the 60 contexts in the subset. We map each context to a key phrase set consisting of all phrases selected by all participants that are associated with that context. We measured the smallest similarity distance between a given phrase and each of the other phrases for that context using the Ratcliff/Obershelp string pattern recognition algorithm[1]. A phrase is trusted to be a validated component of the answer set if it has a similarity score above a certain threshold x out of 1, where 1 would be an exact match, and 0 would contain no overlapping words. After experimentation with threshold values, we selected $x = 0.8$. Duplicate words with similarity above 0.8 are removed from the answer set. After this adjudication process, the average number of key phrases per context is reduced to $K = 5$.

Baseline

Our baseline for the answer generation task uses Named Entity Recognition (NER) to identify successively occurring Dates, Persons, Organizations and Locations as key terms from the context passage.

For example, consider the following sentence:
The Mona Lisa is a 16th-century oil painting created by Leonardo. It's held at the Louvre in Paris.

Using the Stanford NLTK toolkit's NER tagging on the above context yields us the following result:

- *Organization*: Mona/NNP Lisa/NNP
- *Person*: Leonardo/NNP
- *Organization*: Louvre/NNP

Evaluated against human experts

Model	*F1*	*EM*	*Precision*	*Recall*
Baseline	25.94	26.31	30.21	27.11
KPE-Class	24.01	17.67	25.84	22.43
KPE-Gen	**30.59**	**29.91**	**33.58**	**28.10**

Table 1: Results of three key phrase approaches evaluated on ground-truth answers from education experts. The best model for each metric is shown in bold.

- *GPE*: Paris/NNP

We can then identify successively occurring named entities as key terms associated with the context passage. In this case, we identify “Mona Lisa”, “Leonardo”, “Louvre” and “Paris” as the key terms. However, in light of the amount of room for improvement, we tackle the identification of key terms (answers) using deep neural networks.

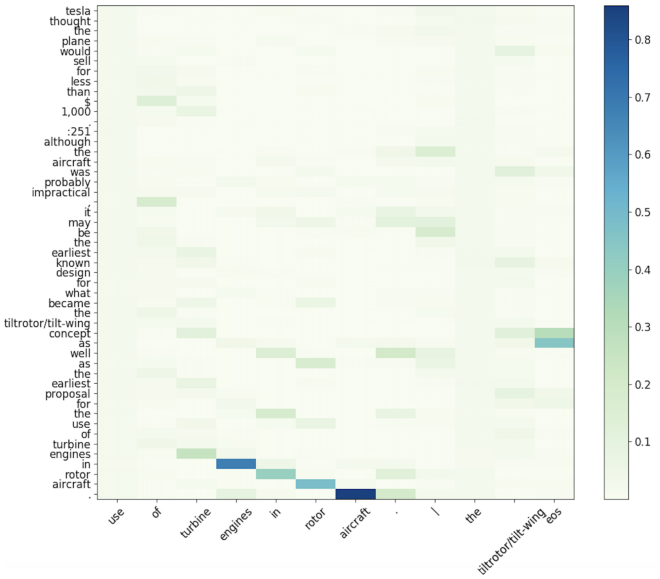


Figure 8: Example attention matrix for KPE-Gen that shows a subset of the input context (y-axis) and two generated key phrases, separated by "|" (x-axis).

Expert Vs. Novice

Here we explore whether there is a significant difference in the performance of our models when tested against our teacher answer set, as opposed to the SQuAD ground truth answers. Table 1 shows the results of evaluating our model on the same 60 context and answer pairs from the human expert dataset, and Table 2 shows the results on SQuAD ground truth. Because SQuAD was generated by Amazon Mechanical Turk annotators with no requirement of relevant educational experience, we consider them as domain novices when it comes to picking the most important phrases for educational purposes. Figure 11 shows an example context from the test set, annotated by both experts and novices, alongside the output of KPE-Gen.

In Tables 1 and 2, EM denotes an *Exact Match* between what was generated by the model, and what was generated by the human. Both Question Generation and Question Answering tasks using SQuAD traditionally leverage a word-level F1 score, however we chose to use a phrase-level F1 calculation. For example, precision is characterized by the number of phrases correctly identified by the model divided by the total number of phrases generated.

Baseline: It was interesting to note that the baseline actually performed higher in matching the human expert key phrases. This points to a larger overlap of Named Entity based answers among experts with baseline answers than there are in the SQuAD answers.

KPE-Class: Though this model does not perform particularly remarkably, we observe that it becomes the most consistent across both datasets by learning a more general distribution. We dive into reasons why this model might not be as competitive in Error Analysis.

KPE-Gen: We find that KPE-Gen is better able to generalize answers in a manner that allows it to perform competitively against both human novice and expert answers. Manual inspection shows that this could be due to another trend in the expert data: When named entities were not the focus of the key phrase, there tended to be more surrounding text used in the answers, and therefore these answers were longer and more complex. In Figure 8, we show an example of the attention matrix KPE-Gen considers when looking at a specific sub-context. KPE-Gen selects the first key phrase to be "use of turbine engines in rotor aircraft" which is also an expert answer; however, the SQuAD ground-truth answers for this sub-context are simply "turbine" or "turbine engines." More examples of KPE-Gen output, compared with our expert dataset and SQuAD, can be seen in Figure 11.

End-To-End Question Generation

In order to validate the usefulness of the key phrases chosen, we generated questions using QG-Net. A randomly selected context is shown in Figure 12 as an example of the end-to-end system. We show the key phrase that is provided in SQuAD alongside two additional key phrases generated by KPE-Gen, and QG-Net is then used to generate associated questions for those key phrases. We also asked an educational domain expert (former teacher) to generate key phrases and questions for this context. We find that KPE-Gen with QG-Net generates question-answer pairs that are more similar to the expert than the pair generated by SQuAD with QG-Net. More rigorous quantitative and qualitative evaluation would help to verify that our end-to-end question generation process indeed generates similar content to educational domain experts.

ERROR ANALYSIS

In this section, we present error analysis to bring more clarity to opportunities for the advancement of key phrase extraction and question generation research.

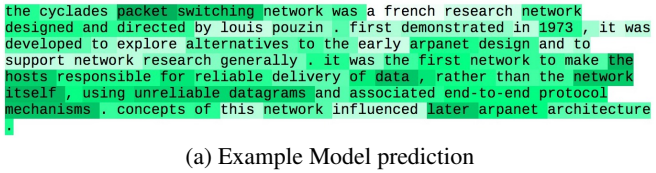
Repeated phrases

The most prevalent error type with the KPE-Gen language model approach is the repetition of phrases. Penalizing the

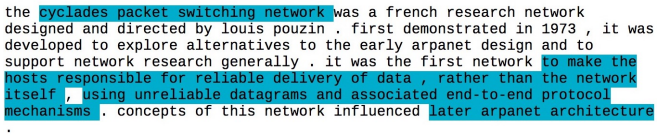
Evaluated against human novices

Model	*F1*	*EM*	*Precision*	*Recall*
Baseline	18.19	26.38	28.79	15.30
KPE-Class	24.62	20.66	29.71	21.02
KPE-Gen	**40.55**	**36.50**	**37.81**	**32.94**

Table 2: Results of three key phrase approaches evaluated on ground-truth answers from SQuAD. The best model for each metric is shown in bold.

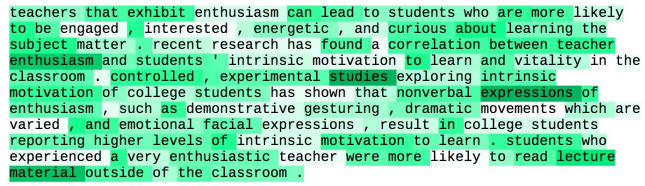


(a) Example Model prediction

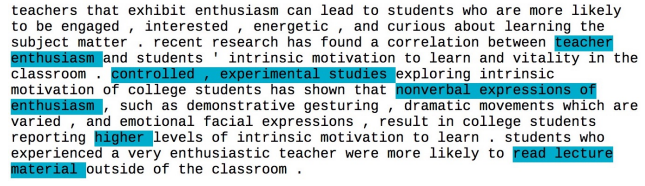


(b) Example ground truth

Figure 9: A visualization of the probability distribution (green) generated by KPE-Class for two examples of SQuAD contexts with their novice ground-truth labels in blue. For the probability distributions, a higher color saturation means a higher probability of being a key phrase.



(a) Example model prediction



(b) Example ground truth novice answers

Figure 10: Second example of predictions by KPE-Class and the corresponding ground-truth for the context.

model for duplicated responses would be something worth considering. Duplicates were easily removed for all approaches during evaluation; however, preventing them from being produced in the first place would likely also lead to more diversity in output (rather than just producing fewer phrases per context).

Contradicting Training Examples

Consider the prediction by the binary classification model illustrated in Figure 9a. The corresponding ground truth for the context is given in Figure 9b. Notice that *later arpanet architecture* is a valid ground truth, whereas *early arpanet design* is not. Our model detects both *later arpanet architecture* and *early arpanet design* as potential answer candidates because both these phrases have similar POS and NER tags associated with them. The POS tags, in particular, are: early/JJ, arpanet/NN, design/NN, later/RB, arpanet/NN, architecture/NN. These strikingly similar examples however are labeled differently. We hypothesize that the model is too simple to learn these nuances.

Effect of Recurring Patterns

Consider the example illustrated using Figures 10a and 10b. There are certain recurring patterns that are very common in usual phrases. These phrases might not be key concepts in the context passage but can bear similar POS (or NER) tag structure to an actual answer (in SQuAD). We present one such instance below.

The ground truth phrase *teacher enthusiasm* has POS structure teacher/NN, enthusiasm/NN. Also, note how the phrase *lecture material* also has the same POS structure: lecture/NN, material/NN. Although the model detects *read lecture material* as keywords, the model is relatively more confident about the phrase *lecture material* than the phrase *read lecture material*, due to the recurring POS pattern (NN, NN) across the examples in SQuAD. Hence, we observe that POS (and NER) information is not always sufficient to determine the kind of answers presented in SQuAD. We might also want to note that the POS (and NER) pattern of the answers in SQuAD is very diverse for the model to have a high F_1 score.

FUTURE WORK

Our work opens a door to future opportunities in generating educational content at scale. There are four particularly interesting directions to explore based upon this work.

Deeper Evaluation: In this work, we are mostly interested in the whether the key phrases generated match with pedagogically valuable key phrases selected by domain experts. In reality, we would be more curious about students' knowledge gains after studying the content automatically generated by a model. To answer this question, we need to carefully design pre- and post-study quizzes and perform controlled lab or online studies to measure students' learning gains as in [3].

Subject-Level Modeling and Evaluation: SQuAD is a rich dataset containing Wikipedia articles in the area of history, social science, biology, and so on [14]. We did not separate our modeling and evaluation from different subjects because SQuAD does not currently contain subject labels. In the future, it would be interesting to recruit crowd workers to label a small set of articles. This would allow us to then apply semi-supervised learning techniques to classify all of the SQuAD articles and perform subject-level modeling and evaluation.

Educational Domain Specific Fine-tuning: So far, we have only used our educational expert data for evaluation. Our next

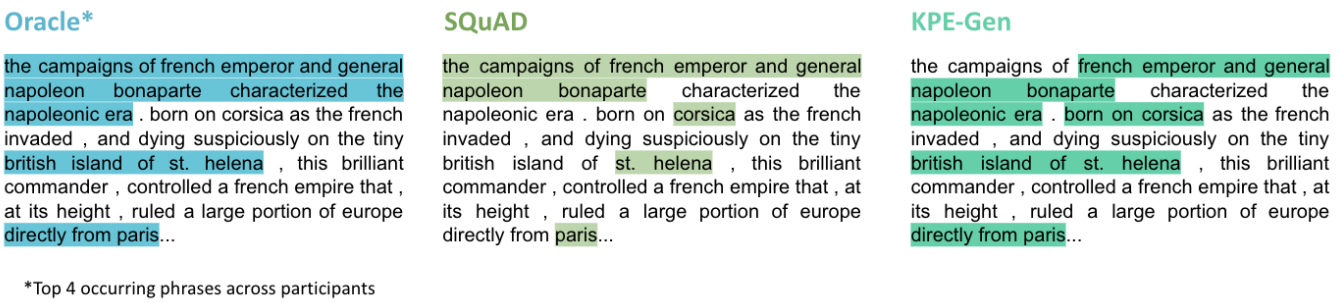


Figure 11: Example phrases obtained from the Oracle (expert dataset), SQuAD (novice dataset) and output from KPE-Gen model.

Context

when rock units are placed under horizontal compression , they shorten and become thicker . because rock units , other than muds , do not significantly change in volume , this is accomplished in two primary ways: through faulting and folding .

Source	Key Phrase	Question Generated
SQuAD (novice)	horizontal compression	**QG-Net:** what are rock units placed under ?
KPE-Gen	when rock units are placed under horizontal compression	**QG-Net:** when do rock units shorten ?
KPE-Gen	faulting	**QG-Net:** along with folding , what is one of the most primary ways of rock units ?
Expert	Through faulting and folding	**Expert:** How do rock units under horizontal compression shorten and become thicker?
Expert	They shorten and become thicker	**Expert:** What happens when rock units are placed under horizontal compression?

Figure 12: The SQuAD dataset provides one answer for the randomly selected context above. KPE-Gen then proposes two additional key phrases which we feed to QG-Net to demonstrate an example of the entire end-to-end system. Expert-generated question-answer pairs are also shown. With KPE-Gen filling the gap between context and question generation, we can create custom assessments from any informational text.

step is to use this data to fine-tune our models and explore whether we can impact the relevance of generated phrases with small amounts of domain-specific data.

Exploring Reinforcement Learning: The models discussed in this paper only learn patterns from the data provided. We aim to explore whether reinforcement learning can provide new insights for text understanding and key phrase extraction.

CONCLUSION

This research is part of a larger strategic effort to advance opportunities for automated educational content generation, including learning modules, quizzes, and interactive smart tutoring systems. To ensure the viability of this research area for educational applications, the technology must be shaped by in-domain interactions and use cases. We introduced two methods for Key Phrase Extraction inspired by and evaluated on the way teachers engage their students in key concept understanding.

Question generation has focused primarily on probing for answers that can be directly extracted from the text. Though SQuAD is purely extractive, by exploring a generative model we take a step towards question generation systems that can be trained with abstractive answer sets, which would be a gateway for question generation techniques, as the research field matures, to develop questions that require deeper reasoning or critical thinking.

Our proposed generative language model, can generalize better to key phrases generated by educational experts than traditionally used word level binary key phrase predictors. Furthermore, we are the first system of its kind to be assessed by domain experts (classroom teachers). Our work bridges educational applications of question generation with our proposed underlying technology by bringing the educator into the evaluation loop. This provides important guidance and insights on generating high-quality questions and answers at scale.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE − 1656518. We thank TAL Education Group for their additional funding support and Google Cloud for academic cloud credits.

DATA ACCESS

To facilitate future research in this area, we will release our expert annotated dataset on request through our website https://hci.stanford.edu/research/smartprimer.

REFERENCES

1. Paul E Black. 2004. Ratcliff/Obershelp pattern recognition. *Dictionary of Algorithms and Data Structures* 17 (2004).
2. Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to Ask: Neural Question Generation for Reading

Comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 1342–1352. DOI: http://dx.doi.org/10.18653/v1/P17-1123

3. Qi Guo, Chinmay Kulkarni, Aniket Kittur, Jeffrey P. Bigham, and Emma Brunskill. 2016. Questimator: Generating Knowledge Assessments for Arbitrary Topics. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI'16)*. AAAI Press, 3726–3732. http://dl.acm.org/citation.cfm?id=3061053.3061140
4. Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-Source Toolkit for Neural Machine Translation. In *Proceedings of ACL 2017, System Demonstrations*. Association for Computational Linguistics, 67–72. http://aclweb.org/anthology/P17-4012
5. Alon Lavie and Abhaya Agarwal. 2007. Meteor: An Automatic Metric for MT Evaluation with High Levels of Correlation with Human Judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation (StatMT '07)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 228–231. http://dl.acm.org/citation.cfm?id=1626355.1626389
6. Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*. http://aclweb.org/anthology/W04-1013
7. Ming Liu, Rafael A Calvo, and Vasile Rus. 2012. G-Asks: An intelligent automatic question generation system for academic writing support. *Dialogue & Discourse* 3, 2 (2012), 101–124.
8. Rui Meng, Sanqiang Zhao, Shuguang Han, Daqing He, Peter Brusilovsky, and Yu Chi. 2017. Deep Keyphrase Generation. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (2017). DOI: http://dx.doi.org/10.18653/v1/p17-1054
9. Ruslan Mitkov and Le An Ha. 2003. Computer-aided Generation of Multiple-choice Tests. In *Proceedings of the HLT-NAACL 03 Workshop on Building Educational Applications Using Natural Language Processing - Volume 2 (HLT-NAACL-EDUC '03)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 17–22. DOI:http://dx.doi.org/10.3115/1118894.1118897
10. Nasrin Mostafazadeh, Ishan Misra, Jacob Devlin, Margaret Mitchell, Xiaodong He, and Lucy Vanderwende. 2016. Generating Natural Questions About an Image. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 1802–1813. DOI: http://dx.doi.org/10.18653/v1/P16-1170
11. 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.
12. Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. http://aclweb.org/anthology/P02-1040
13. Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*. 1532–1543. http://www.aclweb.org/anthology/D14-1162
14. Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *EMNLP 2016: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 2383–2392.
15. 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 (INLG '10)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 251–257. http://dl.acm.org/citation.cfm?id=1873738.1873777
16. Iulian Vlad Serban, Alberto García-Durán, Caglar Gulcehre, Sungjin Ahn, Sarath Chandar, Aaron Courville, and Yoshua Bengio. 2016. Generating Factoid Questions With Recurrent Neural Networks: The 30M Factoid Question-Answer Corpus. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 588–598. DOI: http://dx.doi.org/10.18653/v1/P16-1056
17. Sandeep Subramanian, Tong Wang, Xingdi Yuan, Saizheng Zhang, Yoshua Bengio, and Adam Trischler. 2017. Neural Models for Key Phrase Detection and Question Generation. *arXiv preprint arXiv:1706.04560* (2017).
18. Zichao Wang, Andrew E. Waters, Andrew S. Lan, Phillip J. Grimaldi, Weili Nie, and Richard G. Baraniuk. 2018. QG-Net: A data-driven question generation model for educational content. In *L@S'18: Proceedings of the fifth annual ACM conference on learning at scale*.
19. Zhilin Yang, Junjie Hu, Ruslan Salakhutdinov, and William Cohen. 2017. Semi-Supervised QA with Generative Domain-Adaptive Nets. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 1040–1050. DOI:http://dx.doi.org/10.18653/v1/P17-1096