DISTRACTOR RE-RANKING FOR AUTOMATIC QUIZ GENERATION

A PREPRINT

Girish Kumar
Stanford University
girishk@stanford.edu

Andy Wang
Stanford University
andy2000@stanford.edu

June 13, 2019

ABSTRACT

Automatic quiz generation tackles the problem of generating questions from free-form texts. An important part of this process is generating good distractors as multiple choice options. Current methods rely on similarity search methods using unsupervised word2vec method. In our project, we leverage existing multiple choice choice questions written in textbooks to add a layer of supervision to existing similarity search methods. First, an unsupervised word2vec model is used to extract an initial list of candidates. Second, a supervised re-ranker, trained on the above-mentioned textbook questions, is used to choose the top-k multiple choice distractors. Three ranking approaches were explored - A point-wise ranking SVM, a list-wise ranking neural network and a ranking Generative Adversarial Network (GAN). We evaluate our methods on the SciQ dataset. Qualitative results show that our models are able to generate good distractors and our quantitative results showed that the list-wise ranking neural net performed the best. Future work will focus on improving the syntactic match of distractors and dealing with cases where the chosen distractors could be correct answers to the question.

1 Introduction

In this project we seek to tackle the complex and interesting problem of question generation for the purpose of enhancing the educational experience of students. Learning through evaluation has been widely known to be a very effective method of assessing a students knowledge and offering helpful insights in targeted learning methods. Learning via online resources ie. Wikipedia, blogs, etc has become a growing trend in recent years, however, resources to test mastery of learned concepts is severely lacking

There has been a body of work that addresses this gap by using machine learning to generate multiple choice quiz questions for online texts. Naturally, a very important part of this process is generating good distractors (i.e. wrong multiple choice options designed to confuse students). Currently, methods rely on leveraging a word-vector space to perform a similarity search with the correct-answer as a key. However, this method has its shortcomings as the generated questions might not properly replicate good question asking habits or information processing that a teacher for instance might create. Thus, we hope to leverage existing multiple choice questions written in textbooks in order to add a layer of supervision to existing similarity-search methods and improve the quality of distractors selected.

2 Related Work

2.1 Question Generation

Question generation is a very interesting field presently and one that is currently being explored by many researchers. Currently, a team at the Harbin Institute of Technology are developing a two stage encoder-decoder model that reads in relevant words and the answer location and then produces a question focused upon the answer. This work is particularly interesting as it can provide the groundwork to adding more variety in question generation as current traditional models rely strictly on heuristic rules. Our work is related to that of Aggarwal et al. where a weighted sum of lexical, syntactic features were utilised to select sentences, gaps and distractors from informative texts [1]. Shin et. al. used words clustered using topic models and student errors to pick distractors [2].

2.2 Language Processing

Language processing is also a complimentary field of study for question generation. Researchers are looking into creating models that allow mentors to calibrate models based on language they wish to prioritize in question generation. Teams are ultimately hoping to expand these offerings to standardized testing models which rely heavily on question generation in an effort to create more fair tests that can be evaluated quicker. We've examined a lot of papers and work related to these fields of study in order to help us understand better courses of action to create models more effective to our purpose and also to not repeat the studies of completed work.

3 Datasets

3.1 Multiple-Choice Question Source

To train the supervised re-ranking model, we needed a list of human-generated multiple choice questions. For that, we used the SciQ dataset which contains 13679 crowdsourced science exam questions about Physics, Chemistry and Biology, among others. The questions are in multiple-choice format with 4 answer options each, 3 distractors and 1 correct answer.

3.2 Initial Candidate List Generation

The initial candidate list is obtained by doing a similarity search over an unsupervised word2vec model with the correct answer as a key. Since, we are dealing with science questions, we trained a word2vec model on texts pulled from Wikipedia articles on science related topics. This was first done by downloading a high school science textbook and doing TFIDF searches over a wikipedia dump for every 50 sentences.

Our final dataset hence contained examples containing the following features.

1. Question Sentence: string
2. Correct Answer: string
3. Human Generated Distractors: list of 3 strings
4. Distractor from nearest-neighbor search and not contained in the human-generated list: list of 100 strings

4 Methods

In this section, we outline the different methods we explored for the re-ranking problem. Specifically, given a question, q, correct answer a and list of candidate distractors $\mathbf{D} = (d_0, d_1, ..., d_n)$, the model's tended to rank d_i from a human generated example more favorably. For all the following methods, we encode all strings using the pre-trained Universal Sentence Encoder released on TensorFlow-Hub. The output vector size is 512. We denote this representation as $\psi(\cdot) \in \mathbb{R}^{512}$.

4.1 Ranking SVM

Here, we use an SVM to predict a pointwise score given q, a and d_i. Formally, the input to an SVM would be $concat(\psi(q), \psi(a), \psi(d_i))$, and the SVM is trained to predict 1 if d_i is from a human-generated sample and 0 otherwise. At run time, distractors will be ranked according the score outputted by the SVM.

4.2 Neural Dot Product Ranker

Here, we use a neural network to rank human-generated distractors favorably across the entire list of candidates. We transform the examples such that $\mathbf{D}'$ contains only one human-generated distractor, d_h. We then model an approach to obtain $P_i = P(d_i \mid q,\ a)$. As in [3], we estimate P_i using the dot product of two neural network functions, $\mathbf{h}(\cdot)$, $\mathbf{g}(\cdot)$:

$$P_i = P(d_i \mid \mathbf{B},\, q,\, a) \approx \frac{e^{\mathbf{h}(\psi(q);\psi(a))^{\mathrm{T}}\mathbf{g}(d_i,\, \mathbf{D})}}{\sum_{j=0}^{N} e^{\mathbf{h}(\psi(q);\psi(a))^{\mathrm{T}}\mathbf{g}(d_j,\, \mathbf{D})}} \tag{1}$$

We implement the neural-network as simple one-layer, feed-forwards transformation. The softmax function ensures $\sum_{i=0}^{N} P_i = 1$. Separating the model into two networks allows the network to run efficiently on varying sizes of $\mathbf{D}$. This means that for inference, we could simply transform all the word vectors in the unsupervised word vector model with the one-layer transformation. A simple nearest neighbor search could then be performed over this vector space using $\mathbf{h}(\psi(q); \psi(a))$ as the query vector.

4.3 Ranking GAN

In the previous sections, we saw models that approached the problem in two different ways. The neural dot-product ranker focused on predicting relevant distractors given a query $(concat(\psi(q), \psi(a))$, while the SVM ranker focused on predicting relevancy given a question, answer, distractor triplet. In an attempt to unify these two approaches, we looked to IRGANs [4]. The generative retrieval models $p_\theta(d \mid q,\, a)$, trying to select/generate distractors from the candidate pool. The discriminator is a binary classifier $f_\phi(q, a, d)$ that tries to discriminate well-matched question-answer-discriminator tuples from ill-matched ones. The objective of the discriminator is to maximize the log-likelihood of correctly distinguishing the human-crafted and generator-selected distractors. With the observed human-crafted distractors, and the ones selected from the current optimal generative model, the optimimal parameter ϕ for the discriminative retrieval model is given as follows.

$$\phi^* = \underset{\phi}{\operatorname{argmax}} \sum_{i=1}^{N} \mathbb{E}_{d \sim p_{\text{human}}(d|q_i,a_i)}[\log(\sigma(f_\phi(q_i, a_i, d)))] + \mathbb{E}_{d \sim p_{\theta^*}(d|q_i,a_i)}[1 - \log(\sigma(f_\phi(q_i, a_i, d)))] \tag{2}$$

where σ is the sigmoid function.
As for the generator, since the selection of distractors is discrete, a policy gradient is instead used as in [4].

$$\nabla_\theta J^G(q_i, a_i) \simeq \frac{1}{K} \sum_{k=1}^{K} \nabla_\theta \log p_\theta(d_k | q_i, a_i) \log(1 + \exp(f_\phi(d_k, q_i))) \tag{3}$$

where a sampling approximation is performed in which d_k is the k-th distractor sampled from the $p_\theta(d|q_i, a_i)$. The term $\log(1 + \exp(f_\phi(d_k, q_i))$ acts as the reward for the policy.

5 Experiments

5.1 Evaluation Metrics

At inference, we expect our methods to serve three distractors for every question. As such, we chose our evaluation metric to be precision@3, which computes the proportion of relevant distractors in the set of 3 served by a model. "Relevance", for the purposes of our evaluation, will be taken to be whether a distractor appears in the human-crafted list of distractors.

5.2 Implementation Details

5.2.1 Ranking SVM

We implemented a SVM with a RBF Kernel.

5.2.2 Neural Dot Product Ranker

Both the neural network functions **h** and **g** were one-layer neural networks with a hidden size of 512. A learning rate of 0.00001 was used.

5.2.3 Ranking GAN

The discriminator is implemented as a neural network with one hidden layer of size 512 whilst the generator follows the same implementation as the dot product ranker. A learning rate of 0.001 was used.

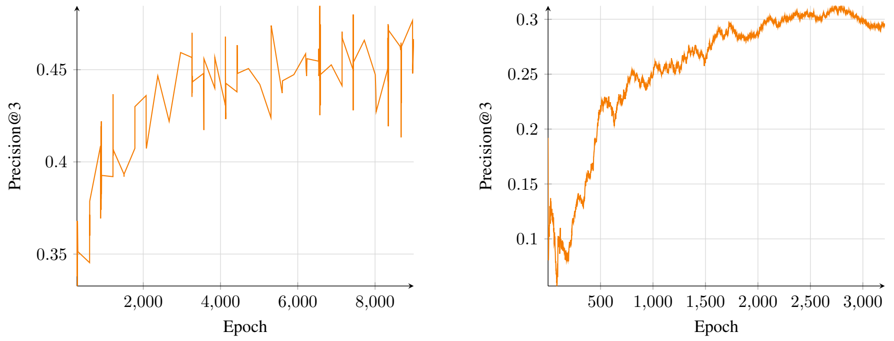

Figure 1: Test Precision@3 for Neural Dot Product Ranker (left) and GAN Ranker (right)

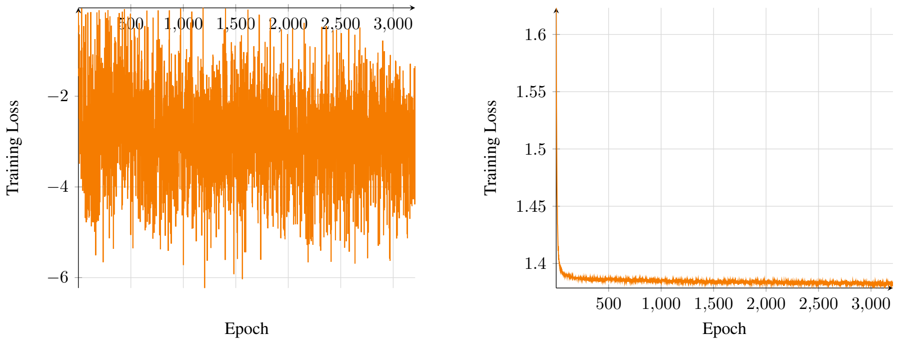

Figure 2: Training Losses for GAN Ranker. Discriminator (left), Generator (right)

5.3 Results

	Baseline Word2Vec	**SVM**	**Neural Ranker**	**Ranking GAN**
Precision@3	0.043	0.314	0.472	0.306

Table 1: Precision@3

The results are clear with the Neural Dot Product yielding the best performance. We conjecture that the neural dot product ranker does better than the SVM primarily because it is directly trained to rank

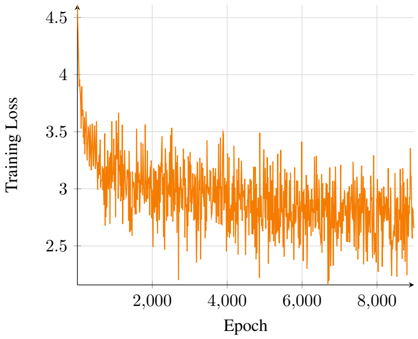

Figure 3: Training Loss for Neural Dot Product Ranker

5.4 Training Curves

Both the GAN and the neural ranker converge on the precision@3 metric. However, it can be observed that in the curves for the GAN, the loss for the generator fluctuates greatly during training. We experimented with different learning rates and optimizers but the phenomenon remained. We attribute this to the general difficulty of training GANs which is complicated by the fact that the generator in our case makes discrete predictions. We leave it to future work to explore the stability of training GANs on discrete data like in our case and expect that improvements would result in a better performance for GANs.

5.5 Qualitative Observations

For the models we trained, the majority of the distractors generated were reasonable, with the neural ranker producing more emantically relevant distractors (as in the case of the second question). Most of the errors we found fell into two categories. First, there were syntactic errors e.g. apostrophes as in question 3. There were also part-of-speech mismatches as in question 6 with *atmospheric*. These problems could possibly be fixed by better data cleaning and by enforcing a part-of-speech-match with a tagger. Another problem was the existence of distractors that could also be correct answers to the question as in question 4 with *cancers* and *leukemia*. This is a much tougher problem and could possibly be solved using an existing ontology such as Freebase.

	Question: What is the major artery of the body, taking oxygenated blood to the organs and muscles of the body? *Answer: aorta*
SVM	Distractor: sinuses Distractor: lungs Distractor: veins
Neural Ranker	Distractor: lungs Distractor: arteries Distractor: heart
GAN	Distractor: sinuses Distractor: drains, Distractor: arteries
	Question: Convex lenses are thicker in the middle than at the edges so they cause rays of light to converge, or meet, at a point called what? *Answer: focus*
SVM	Distractor: the center Distractor: the base Distractor: the apex
Neural Ranker	Distractor: the center Distractor: the base Distractor: the apex
GAN	Distractor: relevance Distractor: ’how’ Distractor: debate
	Question: What is the term for a change in the inherited traits of organisms over time? *Answer: evolution*
SVM	Distractor: speciation Distractor: evolution" . Distractor: macroevolution
Neural Ranker	Distractor: theories Distractor: evolution" Distractor: theory
GAN	Distractor: evolution" Distractor: concept Distractor: generation
	Question: What type of cartilage contains no collagen? *Answer: lamprey cartilage*
SVM	Distractor: joint cartilage Distractor: vertebrae Distractor: limbs
Neural Ranker	Distractor: joint cartilage Distractor: shark cartilage Distractor: fetal cartilage
GAN	Distractor: joint cartilage Distractor: shark cartilage Distractor: fetal cartilage
	Question: A carcinogen is anything in the environment that can cause what disease? *Answer: cancer*
SVM	Distractor: diabetes Distractor: tumors Distractor: tumour
Neural Ranker	Distractor: diabetes Distractor: leukemia Distractor: tumors
GAN	Distractor: cancers Distractor: Cancers Distractor: leukemia
	Question: What is the term for the gases that surround a planet? *Answer: the atmosphere*
SVM	Distractor: evaporation Distractor: planet’s Distractor: emissions
Neural Ranker	Distractor: planet’s Distractor: methane . Distractor: earth’s
GAN	Distractor: methane Distractor: atmospheric Distractor: CO2

Table 2: Trained Model Outputs

6 Conclusion

In this paper, we introduced the problem of supervised re-ranking of candidate distractors using datasets containing actual human-generated questions. Three methods: a pointwise ranking SVM, a listwise ranking neural net and a GAN combining both frameworks were proposed. The listwise neural net yielded the best results. For future work, we would like to explore improving some of the training properties of the GANs. Improving the syntactic form of distractors, along with tackling the problem of distractors being correct answers to the questions are also exciting.

7 Contributions

Andy primarily worked on developing the Ranking SVM model and Girish worked primarily with the Neural Dot and GAN models. The results and analysis were discussed between the both of us. The ultimate report and poster were created together. Code can be found at `https://drive.google.com/drive/folders/11Pueoz3Xs7EEdCfjvzazV-tfS9av7XBB?usp=sharing`

References

[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. Association for Computational Linguistics, 2011.

[2] Jinnie Shin, Qi Guo, and Mark J. Gierl. Multiple-choice item distractor development using topic modeling approaches. *Frontiers in Psychology*, 10:825, 2019.

[3] Girish Kumar, Matthew Henderson, Shannon Chan, Hoang Nguyen, and Lucas Ngoo. Question-answer selection in user to user marketplace conversations. *arXiv preprint arXiv:1802.01766*, 2018.

[4] Jun Wang, Lantao Yu, Weinan Zhang, Yu Gong, Yinghui Xu, Benyou Wang, Peng Zhang, and Dell Zhang. Irgan: A minimax game for unifying generative and discriminative information retrieval models. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 515–524. ACM, 2017.