Question Generation for Creating Exercises and Personalized Feedback in Educational Domain

Devang Kulshreshtha

A thesis submitted to McGill University in partial fulfillment of

the requirements of the degree of

Master of Science

School of Computer Science

McGill University, Montréal

June 2022

Preface

This thesis is submitted in partial fulfillment of the requirements for the degree of Master of Computer Science at McGill University. The research presented here was conducted at the Montreal Institute for Learning Algorithms (Mila), McGill Univerity and Korbit Technologies Inc. under the supervision of Professor Siva Reddy. This work was supported by the MITACS Fellowship, Canada.

The thesis is a collection of two papers, presented in logical order, which include the works I have completed towards question generation for creating question-answering exercises and personalized feedback in educational domain. The papers are preceded by an introductory chapter that relates them to each other and provides background information and motivation for the work, followed by a general discussion and conclusion.

Acknowledgements

I am grateful to many people from Mila and Korbit who have supported me throughout my studies. My research contributions to NLP community would have been impossible without the support of these individuals.

First and foremost, I am endlessly grateful to my advisor Siva Reddy, Professor at Mila/McGill University and Facebook CIFAR AI Chair. His research contributions in NLP inspired me to pursue graduate studies. Transitioning to academia from industry bought a lot of challenges for me, and I learnt major skills observing his way of creating, formulating and solving NLP problems. His intelligent highly-specific feedback over the past two years has greatly accelerated my development as a researcher. I learnt academic writing skills mainly from his feedback on my research manuscripts, and attending his talks at Mila.

I am deeply grateful to my long-term collaborators at Korbit - Iulian Serban (CEO, Korbit Inc) and Ekaterina Kochmar (CSO, Korbit Inc) whose technical expertise and profound insights in

Educational NLP have greatly helped both my research projects at Korbit. I would equally like to thank my co-authors, Muhammad Shayan and Robert Belfer at Korbit, with whom I worked closely in writing papers and deploying models in Korbit production systems. I also thank Stephane Robert, Farid Faraji, Joseph Potochny, and Ariella Smofsky from Korbit engineering team for their help.

I have every member of Prof. Siva Reddy's group to thank for their support. I got wonderful feedback during our group weekly meetings from my friends Zichao Li, Vaibhav Adlakha, Devendra Sachan, Nick Meade, Nathan Schucher, Nouha Dziri, Edoardo Ponti, Benno Krojer, Amirhossein Kazemnejad, Andreas Madsen. I also thank Arinka Jancarik for helping me onboarding to Mila and access to lab resources during my second year.

Abstract

As Intelligent tutoring systems (ITS) such as Coursera, Korbit, Edublog are becoming increasingly popular, a great way for students to learn is to solve QA (question-answering) problems on educational texts. Students also learn by receiving feedback for incorrect answers. Since manual generation of exercises and feedback on large content of educational texts seems infeasible, my thesis work focuses on automatically creating (1) QA exercises from educational resources such as Wikipedia articles (2) personalized feedback for incorrect student answers in an ITS. Our core solution to both problems involves Question Generation (QG) for educational domain.

(1) Q/A Exercise Generation: Existing work on Question Generation relies on supervised labeled data to train Neural Sequence-to-Sequence models. However such supervised data is hard to collect for educational domain, and building Unsupervised Domain Adaptation (UDA) algorithms can circumvent this problem. In our work, we develop *back-training* algorithm which vastly outperforms the popular *self-training* for UDA. Our algorithm significantly reduces the gap between the target domain and synthetic data distribution, and reduces model overfitting to the source domain. We

also release *MLQuestions*[1] dataset to foster research in domain adaptation.

(2) Personalized Feedback Generation: Existing work on generating feedback in an Intelligent Tutoring System (ITS) explores mostly manual, static, and non-personalized feedback. We explore automatically generated questions as personalized feedback in an ITS. Our approach combines cause-effect BERT-similarity based classifier with few-shot Neural Question Generation to generate questions as feedback from missing answer parts. Our model vastly outperforms both simple and strong baselines on student learning gains by 30% when tested on a real dialogue-based ITS[2].

[1] https://github.com/McGill-NLP/MLQuestions
[2] https://www.korbit.ai/

Abrégé

Alors que les systèmes de tutorat intelligents (STI) tels que Coursera, Korbit, Edublog deviennent de plus en plus populaires, un excellent moyen pour les étudiants d'apprendre est de résoudre des problèmes de réponse aux questions, ou "QA" en Anglais sur des textes éducatifs. Les étudiants apprennent également en recevant un retour d'information sur les réponses incorrectes. Étant donné que la création manuelle d'exercices et de commentaires sur un grand nombre de textes éducatifs semble irréalisable, mon travail de thèse se concentre sur la création automatique (1) d'exercices QA à partir de ressources éducatives telles que des articles de Wikipedia (2) de commentaires personnalisés pour les réponses incorrectes des étudiants dans un STI. Notre solution principale à ces deux problèmes implique la génération de questions (QG) pour le domaine éducatif.

(1) Génération d'exercices QA : Les travaux existants sur la génération de questions reposent sur des données supervisées étiquetées pour entraîner des modèles neuronaux de séquence à séquence. Cependant, ces données supervisées sont difficiles à collecter dans le domaine de l'éducation, et la construction d'algorithmes d'adaptation non supervisée de domaine, ou UDA pour "Unsupervised

Domain Adaptation" en Anglais peut contourner ce problème. Dans notre travail, nous développons un algorithme de *back-training* qui surpasse largement *self-training* populaire pour l'UDA. Notre algorithme réduit de manière significative l'écart entre le domaine cible et la distribution des données synthétiques, et réduit la suradaptation du modèle au domaine source. Nous publions également le jeu de données *MLQuestions*[3] pour encourager la recherche sur l'adaptation au domaine.

(2) Génération de rétroaction personnalisé: Les travaux existants sur la génération de rétroaction dans un système de tutorat intelligent (ITS) explorent principalement le rétroaction manuel, statique et non-personnalisé. Nous explorons les questions générées automatiquement comme rétroaction personnalisé dans un ITS. Notre approche combine un classificateur de cause à effet basé sur le score de similarité généré par des modèles BERT, avec la génération neuronale de questions utilisant une approche à quelques coups, pour générer des questions comme rétroaction, à partir de parties de réponses manquantes. Notre modèle surpasse largement les approches de bases et même les plus avancées en termes de gains d'apprentissage des étudiants de jusqu'à 30% lorsqu'il est testé sur un véritable ITS[4] basé sur le dialogue.

[3] https://github.com/McGill-NLP/MLQuestions
[4] https://www.korbit.ai/

List of Acronyms

AI	Artifical Intelligence.
DPR	Dense Passage Retrieval.
IR	Information Retrieval.
ITS	Intelligent Tutoring Systems.
LSTM	Long Short Term Memory.
MAE	Mean Absolute Error.
ML	Machine Learning.
MSE	Mean Squared Error.
NER	Named Entity Recoginition.
NLG	Natural Language Generation.
NLP	Natural Language Processing.
NMT	Neural Machine Translation.
NN	Neural Network.
NQG	Neural Question Generation.

OOD Out of Distribution.

QA Question Answering.

QG Question Generation.

SVM Support Vector Machine.

UDA Unsupervised Domain Adaptation.

List of Papers

"Paper I: Back-Training excels Self-Training at Unsupervised Domain Adaptation of Question Generation and Passage Retrieval". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7064–7078. DOI: 10.18653/v1/2021.emnlp-main.566

"Paper II: Few-shot Question Generation for Personalized Feedback in Intelligent Tutoring Systems". In: *Proceedings of the 11th International Conference on Prestigious Applications of Intelligent Systems, PAIS 2022*.

Contribution of Authors

Devang Kulshreshtha School of Computer Science, Mila/McGill University.

- **Paper I:** First and corresponding author: proposed the idea of collecting MLQuestions dataset, proposed back-training algorithm for Unsupervised Domain Adaptation. Conducted all experiments from data collection to model training and deploying in Korbit ITS. He is the primary writer of the paper.
- **Paper II:** First and corresponding author: proposed error classification of student answers, proposed training Question Generation system as personalized hints. Conducted all experiments from data analysis to model training, formulating the A/B test and deploying in Korbit ITS. He is the primary writer of the paper.

Robert Belfer Korbit Inc Montreal, Canada.

- **Paper I, II:** Co-author: attended weekly sessions to discuss the projects. Contributed to annotating questions quality of Question Generation models in both papers.

Muhammad Shayan Korbit Inc Montreal, Canada.

- **Paper II:** Co-author: Led the integration and testing of hint generation A/B test in production system. Helped in editing the paper.

Iulian Vlad Serban Korbit Inc Montreal, Canada.

- **Paper I, II:** Project Supervisor; contributed to the planning and funding of the study, attended weekly sessions to discuss the projects. Contributed to annotating questions quality of Question Generation models in both papers. Helped in editing the paper.

Ekaterina Kochmar Korbit Inc Montreal, Canada.

- **Paper II:** Project Supervisor; contributed to the planning of the study, attended weekly sessions to discuss the projects. Contributed to annotating questions quality of Question Generation model. Helped in editing the paper.

Siva Reddy School of Computer Science, Mila/McGill University.

- **Paper I, II:** Project Supervisor; led the development, funding, and revision of the studies, weekly discussions, and paper editing.

Contents

Preface i

Acknowledgements ii

Abstract iv

Abrégé vi

List of Abbreviations ix

List of Papers x

Contribution of Authors xi

Table of Contents xv

List of Figures xvii

List of Tables xxi

1 Introduction **1**

1.1 Summary of Papers 6

2 Background & Literature Review **8**

2.1 Question Generation 8

2.2 Question Generation for Educational ITS 11

Papers 21

3 Paper I: Back-Training excels Self-Training at Unsupervised Domain Adaptation of Question Generation and Passage Retrieval 22

3.1 Introduction 24

3.2 Background 27

3.3 Transfer from Source to Target Domain without Adaptation 31

3.4 Unsupervised Domain Adaptation 32

3.5 Domain Adaptation Evaluation 36

3.6 Related Work 43

3.7 Conclusion and Future Work 44

3.8 Acknowledgments 45

References 45

Appendices 54

3.A Appendix 54

3.B Reproducibility Checklist 59

4 Paper II: Few-shot Question Generation for Personalized Feedback in Intelligent Tutoring Systems 65

4.1 Introduction 67
4.2 Background: Exercises in Korbit ITS 69
4.3 Personalized Feedback Generation Model 70
4.4 Experimental Results 78
4.5 Improving Generative Question Answering using Feedback Intervention 82
4.6 Related Work 85
4.7 Conclusion and Future Work 86
References 87

Appendices **95**

4.A Question Generation Model Training Details 95
4.B Cause-Effect Relation Extraction 96

5 Conclusion **98**

5.1 Contribution to Original Knowledge 98
5.2 Limitations and Future Directions 100
Bibliography 121

List of Figures

1.1 An example of an interaction on the platform between the AI tutor, and a student. . 2

2.1 Neural Question Generation proposed by Du et al. [28]. Image taken from the ACL 2017 poster of the same paper. 10

2.2 An example of non-personalized feedback on the platform between the AI tutor, and a student. 19

2.3 An example of personalized feedback on the platform between the AI tutor, and a student. 20

3.1 IID/OOD generalization gaps for Question Generation and Passage Retrieval due to distributional shift between source and target domains. For a fair comparison, the number of candidate passages for IR are kept similar for all datasets. 32

3.2 Self-training and Back-training for UDA. 36

3.3 Evolution of QG model perplexity (PPL) and IR model loss for Self-training vs Back-training as training proceeds on MLQuestions. *Trajectories run from right to left as training loss decreases with time*. Rightmost points are plotted after first mini-batch training, and subsequent points are plotted after each mini-batch training. 37

3.4 DPR embedding similarity scores and QG Log-likelihood scores distribution on MLQuestions synthetic data computed using θ_R and θ_G respectively. 38

3.5 Confusion matrix of actual (row) vs model generated question (column) classes for 100 questions sampled from MLQuestions test set. Classes are abbreviated as Description (D) Comparison (C), Explanation (E), Method (M), and Preference (P). Values are in % where each row sums to 100%. 42

3.A.1 Test set Confusion matrix of Out-of-domain (OOD) and In-domain classes for classifier probability threshold of 0.8. 57

3.A.2 Precision-Recall curve for Test set of 150 questions. AP denotes average precision. 57

4.3.1 An overview of our personalized feedback generation system: (a) Student solution is classified into its error type using cause-effect extractor and BERT similarity. (b) A few-shot QG model generates question from the *cause* of reference solution. (c) Personalized hint is generated using different feedback templates. 71

4.3.2 Illustrating various types of student errors for a *cause-effect* exercise in Korbit ITS. 72

4.4.1 Comparing question quality of T5 with BART based on annotated 80 questions. . . 78

List of Tables

1.1 Examples of questions generated by the model trained on generic dataset such as NaturalQuestions [62] and tested on machine learning domain. *Human* refers to the ideal (gold) question generated by a domain expert. 4

2.1 Examples of document-question pairs. 14

3.1 *Self-Training* and *Back-Training* for unsupervised domain adaptation of question generation and passage retrieval. In self-training, inputs are sampled from the target domain data distribution $P_{\mathscr{T}}$ and their corresponding outputs are generated using a supervised model $P_{\mathscr{S}}$ trained on the source domain. In back-training, the inverse happens: outputs are sampled from $P_{\mathscr{T}}$ and their corresponding inputs are generated using $P_{\mathscr{S}}$. Notation: q and p denote questions and passages respectively, $.\_u$ denotes samples from the target domain and $\hat{.}$ denotes the samples generated by a supervised model trained on the source domain. 24

3.2 Classification of 200 random questions from NaturalQuestions and MLQuestions as per Nielsen [32]. 27

3.3 Notations used throughout the paper. 33

3.4 Results of unsupervised domain adaptation. *No-adaptation* denotes the model trained on NaturalQuestions and tested directly on MLQuestions/PubMedQA without any domain adaptation. 34

3.5 Effect of using consistency filters on Self-Training and Back-Training for MLQuestions. 39

3.6 Evolution of model performance on MLQuestions with increasing iterations: Blue numbers denote increases in performance, while Red numbers denote decrease in performance. 40

3.7 Human evaluations scores between 0-1 on 50 model generated questions for four criteria: Naturalness (N), Coverage (C), Factual Correctness (FC), and Answerability (A). 41

3.8 Examples of generated questions from MLQuestions (first row) and PubMedQA (second row). ST and BT refer to Self-training and Back-training models respectively. 41

3.A.1 Threshold values for different consistency filters. Values are chosen as the third quartile (Q3) of score distribution of synthetic data, accepting 75% of synthetic data for model training. 56

3.B.1 Runtime (in minutes) for each step in domain adaptation models for MLQuestions dataset. Since there are 35K unaligned questions and 50K unaligned passages, a step has different execution times depending on type of training (self/back) or consistency filter (self/cross). 61

3.B.2 Validation set results of unsupervised domain adaptation. *No-adaptation* denotes the model trained on NaturalQuestions and evaluated directly on MLQuestions/PubMedQA dev sets without any domain adaptation. 62

3.B.3 Effect of using consistency filters on Self-Training and Back-Training for MLQuestions validation set. 62

3.B.4 Evolution of model performance on MLQuestions validation set with increasing iterations: Blue numbers denote increases in performance, while Red numbers denote decrease in performance. 63

4.1.1 Non-Personalized vs Personalized Feedback Generation in Korbit ITS. The Personalized Feedback pinpoints correct and missing parts in the answer and provides suggestions on how to improve it. In this case, the student forgot to provide reasoning for their answer and is asked a question about the missing part. . 67

4.2.1 Decomposition of reference solutions in Korbit ITS into their cause and effect. . . . 70

4.3.1 Taxonomy of questions written by annotators and corresponding scores used for question re-ranking. 73

4.4.1 Results of Question Generation Models on standard language evaluation metrics. . 78

4.4.2 Results of Question Re-ranking. 80

4.4.3 Student learning gains on the Korbit ITS at 95% confidence intervals. 82

4.5.1 Results on improving Generative Question Answering Using Hint Intervention . . 85

4.B.1 Examples of the designed cause-effect patterns [4]. 96

Chapter 1

Introduction

Intelligent tutoring systems (ITS) such as Coursera, Korbit, Edublog provide a massive number of students with access to learning on various subjects, having the potential to revolutionize education [37, 56]. In these platforms, students learn by watching video lectures and solving Question Answering (QA) problems on educational texts. Students also learn by getting answers to their questions. Such QA exercises in an ITS can also incorporate giving personalized feedback and explanations in the form of conversation to correct students' answers, thereby engaging students into active and problem solving exercises [2, 22].

Consider the example shown in Figure 1.1 illustrating an interactive dialogue between an AI tutor and a student. First the student is presented with an exercise question whereupon the student attempts to solve the exercise. Their solution is compared against an NLP driven solution checker. If the solution is incorrect, the system responds with a question-based feedback to help students

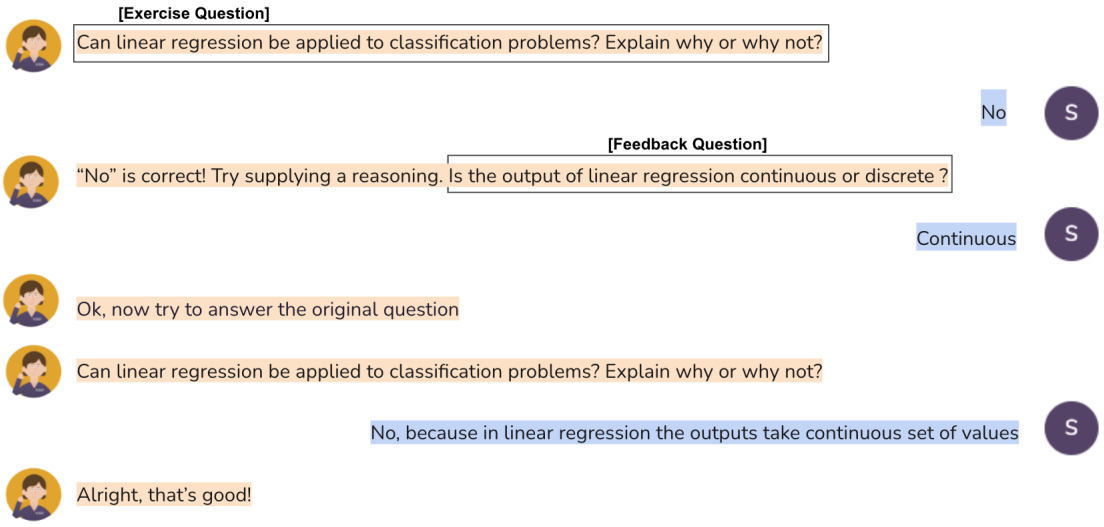

Figure 1.1: An example of an interaction on the platform between the AI tutor, and a student.

arrive at the correct solution to the problem.

Here we see two types of questions - the *Exercise Question* and the *Feedback-based Question* generated by the AI tutor. We also see that the feedback is highly personalized in pinpointing correct/incorrect components in student answers (*'No' is correct!*), and provides suggestions for improvements (*Try supplying a reasoning*).

Building such a system requires creating questions in the form of exercises and feedback. Manual generation of exercise questions on every educational topic becomes infeasible given the large volume of educational content. The question-based feedback is also practically infeasible since the feedback has to generated in real time for thousands of students simultaneously.

Due to the above limitations, an important goal in AI is to generate natural language questions for the purpose of building automated educational tutoring systems. These automated Question

Generation (QG) systems can be deployed in AI tutors for creating exercises, chatbots for providing personalized feedback, and development of annotated data sets for natural language processing research in reading comprehension and Question Answering (QA).

Previous approaches to automated Question Generation have focused on training neural Seq2Seq models [20, 28, 53, 77, 132] on supervised QA datasets such as SQuAD [88] and NaturalQuestions [62]. These datasets are however not targeted towards a specific domain such as education. Due to this, models trained on such datasets cannot generate good questions in our domain of interest i.e. education. Table 1.1 shows output of a QG model trained on generic domain such as Wikipedia articles and questions. We can see that these models have difficulty understanding vocabulary of the test domain (Machine Learning domain in this case). Using these pre-trained models directly without adapting to the target domain often leads to poor generalization due to distributional shift [131]. In domains such as education and medicine, collecting labeled data for tasks like question generation requires domain experts. This motivates the need for Unsupervised Domain Adaptation (UDA) [89] in Question Generation. In UDA, models are further trained on cheap synthetically generated labeled data by exploiting unlabeled data from target domain [89].

Even if we build models that can perform well on target domain, there do not exist any datasets for evaluating question quality for educational domain. This further motivates the need for creating datasets for evaluating domain adaptation models in QG.

Another important component in ITS is the personalization of these questions conditioned on the previous conversation history of students. For example in Figure 1.1 we see that the question-

Passage	Question
Linear regression is used to predict the continuous dependent variable using a given set of independent variables. Logistic Regression is used to predict the categorical dependent variable using a given set of independent variables.	*Model:* What is the use of linear regression in regression? *Human:* What is the difference between linear regression and logistic regression?
Instead of stacking the data, the Convolution Autoencoders keep the spatial information of the input image data as they are, and extract information gently in what is called the Convolution layer.	*Model:* what is the purpose of convolution in image processing? *Human:* How do convolutional autoencoders work?

Table 1.1: Examples of questions generated by the model trained on generic dataset such as NaturalQuestions [62] and tested on machine learning domain. *Human* refers to the ideal (gold) question generated by a domain expert.

based full feedback is highly personalized in detecting the missing/incorrect components in student answer, and then generating question to improve student deficiencies. Previous work for generating automated hints in ITS is non-personalized, relying heavily on manual labour, and expert linguistic knowledge [34, 84]. This motivates the need for personalized feedback generation in ITS.

On the various grounds discussed above, I hypothesize that for personalizing education through AI tutor, progress needs to be made on following three fronts:

1. **Resources for Educational QG**: To develop new datasets for training and evaluation of Question Generation models for education domain. Such datasets can also be used to foster research in Unsupervised Domain Adaptation (UDA).

2. **Algorithms/Models for Educational QG**: To develop algorithms for Question Generation that can generalize/perform well to educational domain.

3. **Personalized Feedback Generation Models**: To build personalized question-based feedback

systems in ITS. The goal here is to pinpoint correct and incorrect/missing components in student answers, and provide feedback in the form of natural language questions that guide students towards improving their answer.

In the following thesis, I present in the works that I have completed towards each of the three components presented above, and show how their culmination contributes to progress in the field. Specifically, my thesis work addresses above three problems by following contributions -

1. **Resources for Educational QG**: We release a new domain-adaptation dataset for evaluation of QG models called *MLQuestions*[1] containing 35K unaligned questions, 50K unaligned passages, and 3K aligned question-passage pairs. The dataset consists of questions from Google search queries and passages from Wikipedia pages related to Machine learning domain (Paper I).

2. **Algorithms/Models for Educational QG**: We develop the *back-training* algorithm for UDA of Question Generation and Passage Retrieval which vastly outperforms the popular self-training [124] algorithm by a mean improvement of 7.8 BLEU4 points on generation, and 17.6% top-20 retrieval accuracy on machine learning and medical domain. Our algorithm significantly reduces the gap between the target domain and synthetic data distribution, and reduces model overfitting to the source domain (Paper I).

   Note that a comparison between back-training and self-training will reveal that back-training requires access to both questions and passages for QG, while self-training requires only

[1] https://github.com/McGill-NLP/MLQuestions

ground-truth passages. However for IR, the reverse is true. Hence one algorithm uses more ground-truth data than other depending on the problem.

3. **Personalized Feedback Generation Models**: We develop a few-shot Question Generation model for providing personalized feedback to students in an ITS. Our model can pinpoint correct and incorrect/missing components in student answers, and provide feedback in the form of natural language questions that guide students towards improving their answer. Although currently the personalization comes solely from the context of ongoing conversation and not the history of past interactions with the user, the feedback generated is highly contextual, domain-aware and effectively targets each student's misconceptions and knowledge gaps. Our model vastly outperforms both simple and strong baselines on student learning gains by 30% when tested on a real dialogue-based ITS[2] (Paper II).

1.1 Summary of Papers

Paper I presents work completed towards the development of benchmark dataset for evaluating QG models for educational domain (**Resources for Educational QG**).

It also proposes back-training algorithm for adapting QG models to any specific domain in an unsupervised fashion (**Algorithms/Models for Educational QG**).

Paper II describes a few-shot Question Generation model for providing personalized feedback to

[2]https://www.korbit.ai/

students in an ITS while solving educational Q/A exercises (**Personalized Feedback Generation Models**).

Chapter 2

Background & Literature Review

In the following thesis, I present works that show progress on the three fronts mentioned in the Introduction. However, before I delve into the mechanics of these works, it is critical to discuss a background and literature review the field of Question Generation as a whole, ITS in education, and the role of QG in education. Accordingly, I will also provide past work done in Question Generation on all three fronts i.e. Resources for Educational Domain, Algorithms/Models for domain adaptation, and Personalized Feedback Generation Models.

2.1 Question Generation

Question Generation (QG) aims to "automatically generating questions from various inputs such as raw text, database, or semantic representation" [97]. Here we review work specifically on text input to generate questions. QG is a challenging, complementary task to Question Answering (QA).

While QA requires an understanding of the input passage and the ability to reason over relevant contexts. But QG additionally integrates the challenges of Natural Language Generation (NLG), i.e., generating grammatically and semantically correct questions.

The practical importance of QG is manifold. Some applications of QG are :

- **Education:** Forming good questions are crucial for evaluating students knowledge and stimulating self-learning. QG can generate assessments for course materials [43] or be used as a component in adaptive, intelligent tutoring systems [70].
- **Search Engine:** Automatically asking clarification questions to better understand users' intention [128].
- **Question Answering:** Question Generation can be used to enrich the training corpus for QA [29]. It can also be used to jointly train with QA for multi-task learning [111].

Traditional rule-based methods for QG mainly focused on generating factoid questions from a single sentence or a paragraph. Most of these methods use question templates and apply predefined linguistic rules to transform a declarative sentence into an question templates [96, 98, 99, 100].

Recently the advent of neural Seq2Seq models in Machine Translation [7] have inspired research in applying these methods in QG. Neural Question Generation (NQG) can also be formulated as an NMT problem, where the passage in source language and question becomes the target language. Below we briefly describe the Seq2Seq architecture for NQG, which was first proposed by Du et al. [28].

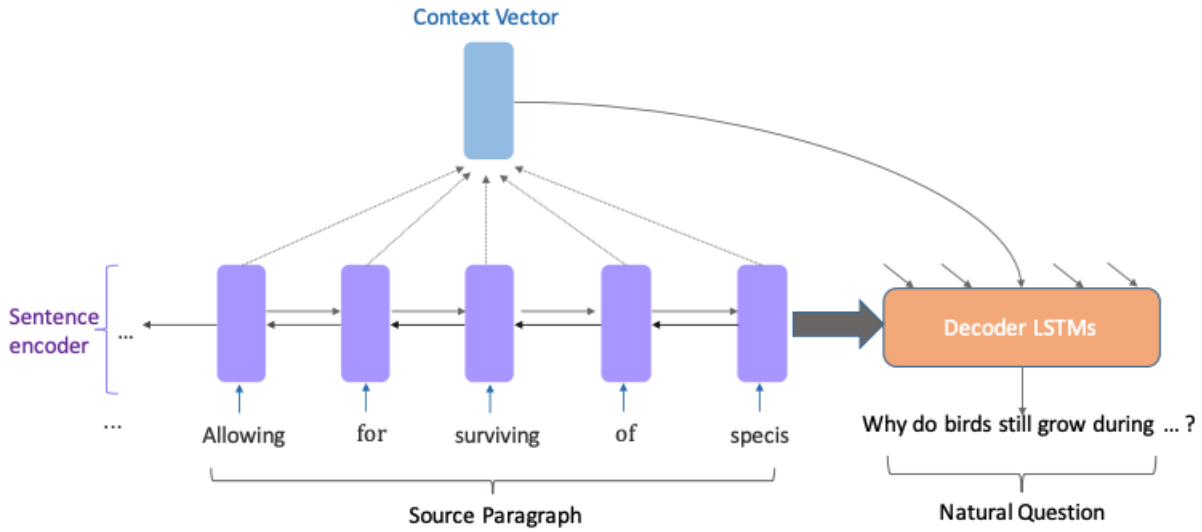

Figure 2.1: Neural Question Generation proposed by Du et al. [28]. Image taken from the ACL 2017 poster of the same paper.

In the Seq2Seq architecture for QG, given a passage $X = \{x_1, x_2, \ldots, x_n\}$ the goal is to generate a question $Y = \{y_1, y_2, \ldots, x_m\}$ from passage X. The model is trained to maximize the likelihood of ground truth conditional probability of question given passage-

$$\hat{Y} = \arg\max_{Y} P(Y|X) \tag{2.1}$$

$$= \arg\max_{Y} \prod_{i=1}^{m} P(y_i|y_{1\ldots i-1}; X) \tag{2.2}$$

Du et al. [28] present the first neural architecture similar to NMT [7] for QG. The architecture is depicted in fig. 2.1 The passage (sentence) is fed into an RNN encoder, and the decoder generates question by outputting one question token at a time. The same attention mechanism as proposed for NMT in Bahdanau et al. [7] is applied on the decoder to pay attention to different parts of sentence

while generating the question. Subsequent works use the same architecture but incorporate other inductive biases in NQG training such as

- **Copying Mechanism:** The copying mechanism proposed originally for Neural Machine Translation [40] enables directly copying relevant input words into the output sentence during decoding. Many NQG models adapt it to copy passage words directly to question such as Harrison and Walker [41], Wang et al. [119], Yuan et al. [127]. This works especially for factoid questions, since it is difficult for the RNN decoder to produce rare words on it's own.

- **Linguistic Features:** Many approaches show that adding additional linguistic features about input passage improve the QG performance. Some examples include adding NER tags [119], co-reference information [41], dependency trees [59] by concatenating these features with input word embedding layer of encoder.

- **Reinforcement Learning:** Many works in NQG directly optimize certain metrics to improve generation quality of questions using policy gradient optimization [110]. Some examples include optimizing BLEU scores [60], semantic similarity between question and passage [61], question fluency measured by a language model [123] etc.

2.2 Question Generation for Educational ITS

In education, questions are one of the most important tools not only in assessing students' knowledge but also in improving their learning [86]. Previous research in linguistics has shown that the number

of questions solved by learners correlates highly with the amount of knowledge retention of relevant subject [8]. With this motivation, many ITS such as Coursera, Edublog, Korbit etc. have various Question Answering exercises at the end of each content unit. Such QA exercises in an ITS may also incorporate giving personalized feedback and explanations in the form of conversation, thereby engaging students into active learning and problem solving exercise [2, 22]. However, manual generation of exercise questions on every educational topic becomes infeasible given the large volume of educational content. Due to the above limitations, automated Question Generation (QG) systems can be useful for building educational resources. As described in 1, progress on research in QG for education has to made in following three fronts-

- Resources for Educational QG
- Algorithms/Models for Educational QG
- Personalized Feedback Generation Models

In the following subsections, I describe the literature review on each of these fronts, and also discuss the limitations of these works, motivating my thesis contributions.

2.2.1 Resources for Educational QG

Most datasets in Question Generation/Answering such as SQuAD [88], NewsQA [112], TriviaQA [48], NarrativeQA [55] etc. are developed for reading comprehension materials. Table 2.1 contains

some examples from each of the above datasets. These datasets are however not suitable for educational content due to the following reasons:

1. **Generic Domain:** Most questions belong to general knowledge and not any subjects in educational domain such as science, math, computer science etc. The topic covered in generic domain contain very few educational subject questions due to limited coverage - the SQuAD dataset for example, was developed by using only 536 Wikipedia articles.

2. **Factoid Questions:** Most questions seek factual detail and the answer can usually be found as a substring of the input passage. While in ITS we want more reasoning-based questions which test higher-level cognitive skills to answer them. A simple heuristic is to count the number of *Why* questions (where the first question-word is *why*). SQuAD contains only 1.2% *Why* questions in its training data.

RACE: Large-scale ReAding Comprehension Dataset From Examinations

The RACE [63] dataset was collected from English examinations designed for middle school and high school students in China. The questions are from educational domain, addressing one of the two shortcomings mentioned above. Though RACE questions are collected in a learning-based context, the questions are mostly factoid (2.1) and test student's understanding of reading comprehension, instead of higher level cognitive skills to answer questions.

Dataset	*Doc*ument-*Q*uestion pairs
SQuAD	*Doc:* ... after Heine German birthplace of Dusseldorf had rejected, allegedly for anti-Semitic motives ... *Q:* Where was Heine born?
RACE	*Doc:* ... There is a big supermarket near Mrs. Green's home. She usually ... *Q:* Where is the supermarket?
TriviaQA	*Doc:* ... is located on natural and man-made barrier islands between the Atlantic Ocean and Biscayne Bay, the latter ... *Q:* Miami Beach in Florida borders which ocean?
LearningQ	*Doc:* ... gases have energy that is proportional to the temperature. The higher the temperature, the higher the energy the gases have... *Q:* If you were given oxygen and hydrogen at the same temperature and pressure, which has more energy?

Table 2.1: Examples of document-question pairs.

LearningQ: A Large-Scale Dataset for Educational Question Generation

Chen et al. [21] create a large-scale Educational QG dataset from KhanAcademy and TED-Ed data sources as a learning and assessment tools for students. However majority (96%) of questions are *clarification* questions asked by students (learners) on the KhanAcademy platform to clarify about an existing concept. Clarification question themselves are not directly useful for testing student's learning abilities for higher-level cognitive skills.

In my thesis, I will propose a dataset for educational domain in the first paper which addresses both limitations discussed above.

2.2.2 Algorithms/Models for Educational QG

Educational QG models can be built by either training Seq2Seq models on educational QG datasets, or performing domain adaptation of QG model trained on generic domain. Existing research is mostly limited on both methods, and is described below.

Training models on Educational QG datasets

Though the RACE dataset contains educational questions, Lai et al. [63] focus on training Question Answering models, and not QG models.

The only work to the best of our knowledge for educational QG model is by Chen et al. [21]. The authors build three models on LearningQ dataset -

- **H&S Baseline** is a rule-based system [43] to generate multiple questions from source paragraph and pick the best question.
- **NN-based Seq2Seq** is an encoder-decoder framework proposed for Neural Machine Translation by Sutskever et al. [109]. The passage (sentence) is fed into an RNN encoder, and the decoder generates question by outputting one question token at a time.
- **Attention Seq2Seq** is similar to previous model, with the added attention mechanism also proposed for NMT by Bahdanau et al. [7]. The attention mechanism allows focusing on relevant information in the source passage and generate a question using this information.

Current state-of-the-art NLP neural models are transformers [115] which perform self-attention

over input text. These models are much faster than LSTM-based neural networks because they process input sequence parallelly as opposed to sequential token processing. Shortly after, with the advent of deep bidirectional transformers (e.g., BERT [26], RoBERTa [72], BART [66]) pretrained on a massive amount of data, near-human-level performance has been reached on many NLP tasks. In my thesis papers, I will explore these models for Question Generation in more detail within the papers presented in the thesis.

Domain Adaptation for QG

There is no direct work in our knowledge for domain adaptation in Question Generation. Previous research focuses on creating QG datasets and building supervised algorithms/architectures to train QG models on these datasets. Sachan and Xing [101] apply self-training for training QG model from limited labeled data and unsupervised unlabeled data. This is done by first pretraining QG model on labeled question-passage data, then generating questions from unlabeled passages using QG model to prepare synthetic data, and finally finetuning QG model on passages and synthetic question pairs.

Self-training can also be applied for building educational QG models. First, train a QG model on source (generic domain). Next generate synthetic questions from passages of target (educational domain) using QG model. Finally, fine-tune QG model on synthetic question and passage pairs.

The problem with self-training is that it leads to model overfitting to source domain. This is because in self-training, inputs are sampled from target domain but the outputs are generated from

source domain model. Outputs generated by source domain model leads to overfitting to source domain which is not desirable for domain adaptation, and causes poor model performance which I demonstrate in Paper-I of my thesis. In the same paper, I propose the *back-training* algorithm for domain adaptation of Question Generation and Passage Retrieval where outputs are sampled from target domain to reduce overfitting to source domain. Back-training vastly outperforms the popular self- training algorithm by addressing the overfitting problem.

2.2.3 Personalized Feedback Generation in ITS

Intelligent Tutoring Systems (ITS) are *"computer-based instructional systems with models of instructional content that specify what to teach, and teaching strategies that specify how to teach"* [121]. One of the key strengths of ITS is that they can address personalization in computer-based environments. Multiple previous studies demonstrate that personalized human tutoring helps students achieve their learning goals effectively [4, 13, 17, 44, 46], since it allows a tutor to understand the effective state of the student and provide personalized feedback by adapting instructions accordingly. However, one-on-one tutoring is generally seen as too costly to be conducted on a large scale in most societies, whereas ITS are a low-cost alternative to human tutors as they can provide personalized tutoring [5, 82]. Previous research confirms that personalization in ITS leads to substantially higher learning outcomes for students [105, 106].

One such example of such a system is Korbit [1], a large-scale AI-powered personalized ITS.

[1] https://www.korbit.ai/

Students watch video lectures on data science topics and working on problem-solving exercises created by domain experts. While going through exercises, the student's answers are compared to reference solutions using an ML-based solution checker. At this point if student answer is marked as incorrect, a feedback/hint is shown to the student to guide them towards the correct answer.

Many ITS however rely heavily on expert hand-crafted rules to generate feedback hints which becomes infeasible for large amounts of educational texts. An important research goal is to thus develop automated feedback systems from student-tutor conversation history [74, 83]. Existing work mainly focuses on non-personalized hints created using template-based methods [12, 71]. However, students make various type of mistakes (such as grammatical errors, correct answers with incorrect reasoning, and so on), and showing the same hint to address different mistakes is not efficient in improving students' answers, and might even further confuse them. As a result, this can lead to lower motivation and a decrease in the overall study time spent on an ITS platform.

Figure 2.2 demonstrates an example interaction of student with Korbit ITS. Consider the case where student supplies the correct answer (*"I think it's a classification task"*) without an explanation. The feedback shown by the AI tutor is non-personalized. The model produces a generic hint irrespective of any student answer, saying *'Thats not right'* even though the main answer is correct. This further confuses the student and causes them to change their correct answer in the next attempt.

Hence, an important component in ITS is the personalization of these feedbacks conditioned on the previous conversation history of students. For example in fig. 2.3 we see that the question-based

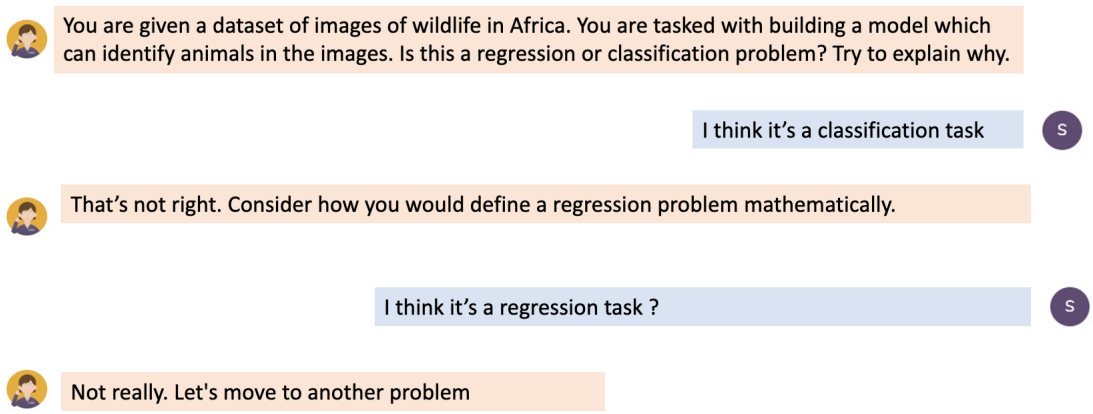

Figure 2.2: An example of non-personalized feedback on the platform between the AI tutor, and a student.

feedback is highly personalized in detecting the missing/incorrect components in student answer (*"No" is correct!*), and prompts them to supply an explanation (*Try supplying a reasoning*). It then asks a clarifying question steering the student towards the reasoning part which was missing (*Is the output of linear regression continuous or discrete?*). As a result, the student is able to provide a correct solution.

In my Paper-II I will present an NLP model based on few-shot Question Generation for generating personalized feedback in ITS. We will see how applying leveraging Question Generation to generate personalized feedback improves student learning gains, and their interaction time with ITS platforms such as Korbit.

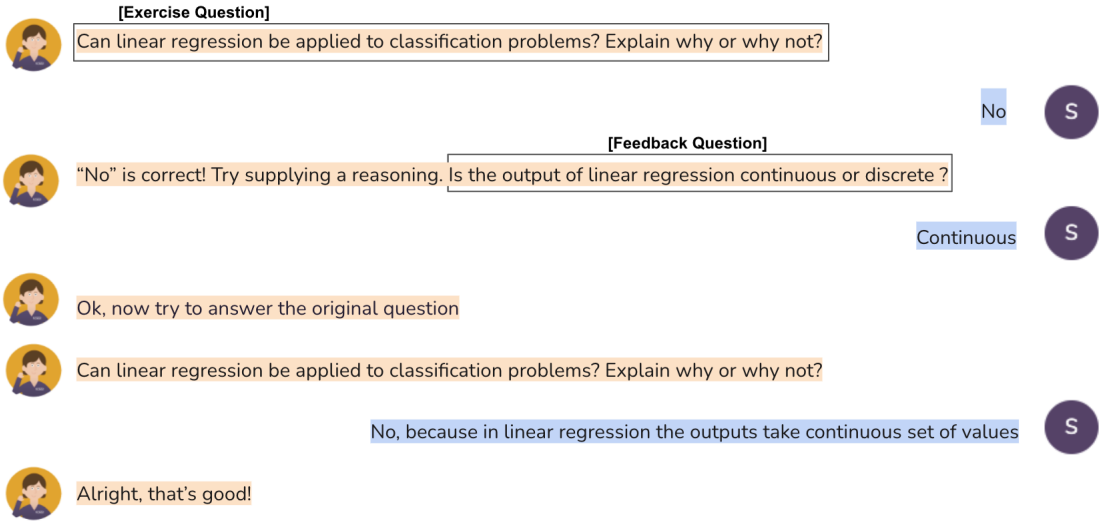

Figure 2.3: An example of personalized feedback on the platform between the AI tutor, and a student.

Papers

Chapter 3

Paper I: Back-Training excels Self-Training at Unsupervised Domain Adaptation of Question Generation and Passage Retrieval

Devang Kulshreshtha, Robert Belfer, Iulian Vlad Serban, Siva Reddy

Published in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7064–7078. DOI: 10.18653/v1/2021.emnlp-main.566

Problem Statement: The popular self-training algorithm for Unsupervised Domain Adaptation (UDA) generates synthetic training data where natural inputs are aligned with noisy outputs. This leads to larger significant gap between the target domain and synthetic data distribution, and increases model overfitting to the source domain. We propose *back-training* algorithm to address above problems which generates synthetic data aligning natural outputs with noisy inputs. Back-training vastly outperforms self-training by a mean improvement of 7.8 BLEU-4 points on generation, and 17.6% top-20 retrieval accuracy across both domains. We also release a new domain-adaptation dataset- *MLQuestions* to evaluate QG models for educational domain.

Abstract

In this work, we introduce back-training, an alternative to self-training for unsupervised domain adaptation (UDA) from source to target domain. While self-training generates synthetic training data where natural inputs are aligned with noisy outputs, back-training results in natural outputs aligned with noisy inputs. This significantly reduces the gap between the target domain and synthetic data distribution, and reduces model overfitting to the source domain. We run UDA experiments on question generation and passage retrieval from the *Natural Questions* domain to machine learning and biomedical domains. We find that back-training vastly outperforms self-training by a mean improvement of 7.8 BLEU-4 points on generation, and 17.6% top-20 retrieval accuracy across both domains. We further propose consistency filters to remove low-quality synthetic data before training. We also release a new domain-adaptation dataset- *MLQuestions* containing 35K unaligned questions, 50K unaligned passages, and 3K aligned question-passage pairs.

Algorithm	Synthetic Training Data Input	Output
Question Generation (QG)		
Self-Training	$p_u \sim P_{\mathscr{T}}(p)$	$\hat{q} \sim P_{\mathscr{S}}(q\|p_u)$
Back-Training	$\hat{p} \sim P_{\mathscr{S}}(p\|q_u)$	$q_u \sim P_{\mathscr{T}}(q)$
Passage Retrieval (IR)		
Self-Training	$q_u \sim P_{\mathscr{T}}(q)$	$\hat{p} \sim P_{\mathscr{S}}(p\|q_u)$
Back-Training	$\hat{q} \sim P_{\mathscr{S}}(q\|p_u)$	$p_u \sim P_{\mathscr{T}}(p)$

Table 3.1: *Self-Training* and *Back-Training* for unsupervised domain adaptation of question generation and passage retrieval. In self-training, inputs are sampled from the target domain data distribution $P_{\mathscr{T}}$ and their corresponding outputs are generated using a supervised model $P_{\mathscr{S}}$ trained on the source domain. In back-training, the inverse happens: outputs are sampled from $P_{\mathscr{T}}$ and their corresponding inputs are generated using $P_{\mathscr{S}}$. Notation: q and p denote questions and passages respectively, $._u$ denotes samples from the target domain and $\hat{.}$ denotes the samples generated by a supervised model trained on the source domain.

3.1 Introduction

In domains such as education and medicine, collecting labeled data for tasks like question answering and generation requires domain experts, thereby making it expensive to build supervised models. Transfer learning can circumvent this limitation by exploiting models trained on other domains where labeled data is readily available [4, 39]. However, using these pre-trained models directly without adapting to the target domain often leads to poor generalization due to distributional shift [53]. To address this issue, these models are further trained on cheap synthetically generated labeled data by exploiting unlabeled data from target domain [36]. One such popular data augmentation method for unsupervised domain adaptation (UDA) is *self-training* [50].

In self-training, given a pre-trained model that can perform the task of interest in a source domain and unlabeled data from the target domain, the pre-trained model is used to predict noisy labels for the target domain data. The pre-trained model is then fine-tuned on synthetic data to adapt to the new domain. To improve the quality of the synthetic data, it is also common to filter out low-confidence model predictions [55].

A model fine-tuned on its own confidence predictions might suffer from confirmation bias which leads to overfitting [51]. This means that the distributional gap between the target domain's true output distribution and the learned output distribution could grow wider as training proceeds. In this paper, we propose a new training protocol called *back-training* which closes this gap (the name is inspired from *back-translation* for machine translation). While self-training generates synthetic data where noisy outputs are aligned with quality inputs, back-training generates quality outputs aligned with noisy inputs. The model fine-tuned to predict real target domain outputs from noisy inputs reduces overfitting to the source domain [45], and matches the target domain distribution more closely.

We focus on unsupervised domain adaptation (UDA) of Question Generation (QG) and Passage Retrieval (IR) from generic domains such as Wikipedia to target domains. Our target domain of interest is *machine learning*, as it is a rapidly evolving area of research. QG and IR tasks could empower student learning on MOOCs [18]. For example, from a passage about linear and logistic regression, an education bot could generate questions such as *what is the difference between linear and logistic regression?* to teach a student about these concepts. Moreover, IR models could help

students find relevant passages for a given question [15]. In this domain, unsupervised data such as text passages and questions are easy to obtain separately rather than aligned to each other.

We also perform our main domain adaptation experiments on *biomedical* domain using PubMedQA dataset [19] to further strengthen our hypothesis.

Table 3.1 demonstrates the differences between self-training and back-training for QG and IR. Consider the QG task: for self-training, we first train a supervised model $P_{\mathscr{S}}(q|p)$ on the source domain that can generate a question q given a passage p. We use this model to generate a question $\hat{q}$ for an unsupervised passages p_u sampled from the target domain distribution $P_{\mathscr{T}}(p)$. Note that $\hat{q}$ is generated conditioned on the target domain passage using $P_{\mathscr{S}}(q|p_u)$. We use the pairs (p_u, $\hat{q}$) as the synthetic training data to adapt $P_S(q|p)$ to the target domain. In back-training, we assume access to unsupervised questions and passages from the target domain. We first train an IR model $P_S(p|q)$ on the source domain, then sample a question q_u from the target domain distribution $P_{\mathscr{T}}(q)$. We condition the retriever on this question i.e., $P_S(p|q_u)$, and retrieve a passage $\hat{p}$ from the target domain and treat it as a noisy alignment. We use the pairs ($\hat{p}$, q_u) as the synthetic training data to adapt $P_S(q|p)$. Table 3.1 also describes the details of domain adaptation for the passage retriever.

Our contributions and findings are as follows: 1) We show that QG and IR models trained on NaturalQuestions [23] generalize poorly to target domains, with at least 17% mean performance decline on both QG and IR tasks. 2) Although self-training improves the domain performance marginally, our back-training method outperforms self-training by a mean improvement of 7.8 BLEU-4 points on generation, and 17.6% top-20 retrieval accuracy across both target domains. 3)

Taxonomy	Examples (from MLQuestions)	Description *(Frequent Wh-words)*	Distribution (%) NaturalQuestions	MLQuestions
DESCRIPTION	What is supervised learning with example?	Asking definition or examples about a concept *(What, Who, When, Where)*	86%	39%
METHOD	How do you compute vectors in Word2Vec?	Computational or procedural questions - *(How)*	1%	15%
EXPLANATION	Why does ReLU activation work so surprisingly well?	Causal, justification or goal-oriented questions - *(Why)*	3%	18%
COMPARISON	What is the difference between LDA and PCA?	Ask to compare more than one concept with each other	5%	10%
PREFERENCE	Is language acquisition innate or learned?	Yes/No or select from valid set of options - *(Is, Are)*	5%	18%

Table 3.2: Classification of 200 random questions from NaturalQuestions and MLQuestions as per Nielsen [32].

We further propose consistency filters to remove low-quality synthetic data before training. 4) We release *MLQuestions*: a domain adaptation dataset for the machine learning domain containing 35K unaligned questions, 50K unaligned passages, and 3K aligned question-passage pairs.

3.2 Background

In this section, we describe the source and target domain datasets, models for question generation and passage retrieval, and the evaluation metrics.

3.2.1 Source Domain: NaturalQuestions

We use the NaturalQuestions dataset [23] as our source domain. NaturalQuestions is an open-domain question answering dataset containing questions from Google search engine queries paired with answers from Wikipedia. We use the long form of the answer which corresponds to passages

(paragraphs) of Wikipedia articles. It is the largest dataset available for open-domain QA, comprising of 300K training examples, each example comprising of a question paired with a Wikipedia passage. We label 200 random questions of NaturalQuestions and annotate them into 5 different classes based on the nature of the question as per Nielsen et al. [33]. Table 3.2 shows these classes and their distribution. As seen, 86% of them are descriptive questions starting with *what, who, when* and *where*. Refer to Section 3.A.2 for details on dataset pre-processing and Section 3.A.4 for detailed taxonomy description.

3.2.2 Target Domain I: Machine Learning

Our first target domain of interest is machine learning. There is no large supervised QA dataset for this domain, and it is expensive to create one since it requires domain experts. However, it is relatively cheap to collect a large number of ML articles and questions. We collect ML concepts and passages from the Wikipedia machine learning page[1] and recursively traverse its subcategories. We end up with 1.7K concepts such as *Autoencoder*, *word2vec* etc. and 50K passages related to these concepts.

For question mining, we piggy-back on Google Suggest's *People also ask* feature to collect 104K questions by using above machine learning concept terms as seed queries combined with question terms such as *what*, *why* and *how*. However, many questions could belong to generic domain due to ambiguous terms such as *eager learning*. We employ three domain experts to

[1] `https://en.wikipedia.org/wiki/Category:Machine_learning`

annotate 1000 questions to classify if a question is in-domain or out-of-domain. Using this data, we train a classifier [27] to filter questions that have in-domain probability less than 0.8. This resulted in 46K in-domain questions, and has 92% accuracy upon analysing 100 questions. Of these, we use 35K questions as unsupervised data. See section 3.A.3 for classifier training details and performance validation.

The rest of the 11K questions are used to create supervised data for model evaluation. We use the Google search engine to find answer passages to these questions, resulting around 11K passages. Among these, we select 3K question and passage pairs as the evaluation set for QG (50% validation and 50% test). For IR, we use the full 11K passages as candidate passages for the 3K questions. We call our dataset *MLQuestions*.

Table 3.2 compares MLQuestions with NaturalQuestions. We note that MLQuestions has higher diversity of question classes than NaturalQuestions, making the transfer setting challenging.

3.2.3 Target Domain II: Biomedical Science

Our second domain of interest is biomedicine for which we use PubMedQA [19] dataset. Questions are extracted from PubMed abstract titles ending with question mark, and passages are the conclusive part of the abstract. As unsupervised data, we utilize PQA-U(nlabeled) subset containing 61.2K unaligned questions and passages. For supervised data, we use PQA-L(abeled) subset of 1K question-passage pairs manually curated by domain experts. We use the same dev-test split of 50-50% as [19] as the evaluation set for QG. For IR, in order to have the same number of candidate

passages as MLQuestions, we combine randomly sampled 10K passages from PQA-U with 1K PQA-L passages to get 11K passages as candidate passages for 1K questions.

3.2.4 Question Generation Model

We use BART [24] to train a supervised QG model on NaturalQuestions. BART is a Transformer encoder-decoder model pretrained to reconstruct original text inputs from noisy text inputs. Essentially for QG, BART is further trained to learn a conditional language model $P_{\mathscr{S}}(q|p)$ that generates a question q given a passage p from the source domain. For experimental details, see 3.A.1.

3.2.5 Passage Retrieval Model

We use the pretrained Dense Passage Retriever (DPR; Karpukhin et al. 20) on NaturalQuestions. DPR encodes a question q and passage p separately using a BERT bi-encoder and is trained to maximize the dot product (similarity) between the encodings $E_P(p)$ and $E_Q(q)$, while minimizing similarity with other closely related but negative passages. Essentially, DPR is a conditional classifier $P_S(p|q)$ that retrieves a relevant passage p given a question q from the source domain. For model training details, see 3.A.1.

3.2.6 Evaluation Metrics

We evaluate question generation using standard language generation metrics: BLEU1-4 [34], METEOR [3] and ROUGE$_L$ [26]. They are abbreviated as B1, B2, B3, B4, M, and R respectively throughout the paper. We also perform human evaluation on the model generated questions. For passage retrieval, we report top-k retrieval accuracy for $k = 1, 10, 20, 40, 100$ following Karpukhin et al. [20] by measuring the fraction of cases where the correct passage lies in the top k retrieved passages. We consider 11K passages in all datasets for retrieval during test time.

3.3 Transfer from Source to Target Domain without Adaptation

We investigate how well models trained on NaturalQuestions transfer directly to our target domains without any domain adaptation. For comparison, we also present the results on NaturalQuestions. To be fair, we sample equal number of samples from the development set of NaturalQuestions as in the test set of MLQuestions and PubMedQA for QG and IR tasks. Figure 3.1 shows the results. We observe high performance drops across all generation metrics (14-20%) from NaturalQuestions (IID data) to MLQuestions and PubMedQA (OOD Data). Human evaluation on QG (see Table 3.7) also reveals that the generated questions are either generic, or fail to understand domain-specific terminology. OOD performance in the IR task is even worse (25-40% drop), revealing a huge distribution shift between the source and target domain.

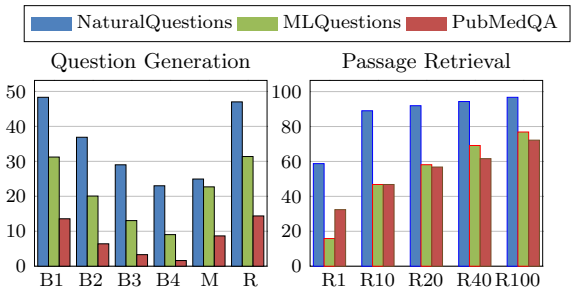

Figure 3.1: IID/OOD generalization gaps for Question Generation and Passage Retrieval due to distributional shift between source and target domains. For a fair comparison, the number of candidate passages for IR are kept similar for all datasets.

3.4 Unsupervised Domain Adaptation

In this section, we describe self-training and back-training methods to generate synthetic training data for unsupervised domain adaptation (UDA). We also introduce consistency filters to further improve the quality of the synthetic data.

3.4.1 Problem Setup

The source domain consists of labeled data containing questions paired with passages $\mathscr{D}_{\mathscr{S}} \equiv \{(q_s^i, p_s^i)\}_{i=1}^{m}$. The target domain consists of unlabeled passages $\mathscr{P}_{\mathscr{U}} \equiv \{p_u^i\}_{i=1}^{m_p}$, and unlabeled questions $\mathscr{Q}_{\mathscr{U}} \equiv \{q_u^i\}_{i=1}^{m_q}$. Note that $\mathscr{P}_{\mathscr{U}}$ and $\mathscr{Q}_{\mathscr{U}}$ are *not necessarily* aligned with each other. Given this setup, our goal is to learn QG and IR models with parameters $\theta \equiv \{\theta_G, \theta_R\}$ that can achieve high generation and retrieval performance on target domain T. Table 3.3 describes the notations used across the paper.

Notation	Definition
S, T	Source, Target Domain
P_S, P_T	Source, Target data distribution
$\mathscr{D}_{\mathscr{S}} \equiv \{(q_s^i, p_s^i)\}_{i=1}^{m}$	Source labeled corpus
$\mathscr{P}_{\mathscr{U}} \equiv \{p_u^i\}_{i=1}^{m_p}$	Target unlabeled passages
$\mathscr{Q}_{\mathscr{U}} \equiv \{q_u^i\}_{i=1}^{m_q}$	Target unlabeled questions
$\theta \equiv \{\theta_G, \theta_R\}$	QG, IR Models
S_G, S_R	Synthetic data for QG, IR

Table 3.3: Notations used throughout the paper.

3.4.2 Self-Training for UDA

Self-training [50] involves training a model on its own predictions. We present the proposed self-training for UDA in Algorithm 1. First the baseline models θ_G and θ_R are trained on the source passage-question corpus $\mathscr{D}_{\mathscr{S}}$. Then, at each iteration, the above models generate pseudo-labeled data from unlabeled passages $\mathscr{P}_{\mathscr{U}}$ for question generation and questions $\mathscr{Q}_{\mathscr{U}}$ for passage retrieval. For QG, θ_G generates a question $\hat{q}$ for each $p_u \in \mathscr{P}_{\mathscr{U}}$ and adds $(p_u, \hat{q})$ to synthetic data S_G. For IR, θ_R retrieves a passage $\hat{p}$ from $\mathscr{P}_{\mathscr{U}}$ for each $q_u \in \mathscr{Q}_{\mathscr{U}}$ and adds $(q_u, \hat{p})$ to S_R. The models θ_G and θ_R are fine-tuned on S_G and S_R respectively. The process is repeated for a desired number of iterations, which we refer to as *iterative refinement*. Note that in self-training, inputs are sampled from target domain and the outputs are predicted (noisy).

3.4.3 Back-Training for UDA

The main idea of back-training is to work backwards: start with true output samples from the target domain, and predict corresponding inputs which aligns the most with the output. While self-training

assumes inputs are sampled from the target domain distribution, back-training assumes outputs are sampled from the target domain distribution. When two tasks are of dual nature (i.e., input of one task becomes the output of another task), back-training can be used to generate synthetic training data of one task using the other, but on a condition that outputs can be sampled from the target domain distribution. QG and IR tasks meet both criteria. For QG, we have unlabeled questions in the target domain and its dual friend IR can retrieve their corresponding input passages from the target domain. For IR, we have passages in the target domain and QG can generate their input questions. Formally, for QG, the IR model θ_R retrieves passage $\hat{p}$ from $\mathscr{P}_{\mathscr{U}}$ for each $q_u \in \mathscr{Q}_{\mathscr{U}}$ and adds $(\hat{p}, q_u)$ to S_G. For IR, the QG model θ_G generates a question $\hat{q}$ for each $p_u \in \mathscr{P}_{\mathscr{U}}$ and adds $(\hat{q}, p_u)$ to S_R.

Similarities with *back-translation* Back-translation is an effective method to improve machine translation using synthetic parallel corpora containing human-produced target language sentences paired with artificial source language translations [13, 42]. Back-training is inspired by this idea, however it is not limited to machine translation.

Dataset	Model	Question Generation						Passage Retrieval			
		B1	B2	B3	B4	M	R	R@1	R@20	R@40	R@100
MLQuestions	No-adaptation	31.23	20.07	13.05	9.04	22.70	31.38	15.86	58.13	69.13	76.86
	Self-Training	31.81	20.74	13.61	9.43	23.31	32.18	17.86	65.26	74.13	83.06
	Back-Training	**44.12**	**32.86**	**24.21**	**18.48**	**23.83**	**43.97**	**24.53**	**77.73**	**84.8**	**91.73**
PubMedQA	No-adaptation	13.57	6.41	3.31	1.62	8.67	14.38	32.4	56.8	61.6	72.2
	Self-Training	13.36	6.28	3.25	1.64	8.84	15.00	32.8	57.0	63.6	72.8
	Back-Training	**26.71**	**17.01**	**11.80**	**8.25**	**16.99**	**25.14**	**55.4**	**79.8**	**81.8**	**85.8**

Table 3.4: Results of unsupervised domain adaptation. *No-adaptation* denotes the model trained on NaturalQuestions and tested directly on MLQuestions/PubMedQA without any domain adaptation.

Algorithm 1 Vanilla Self-Training Back-Training for unsupervised domain adaptation. Vanilla algorithms can be improved further using consistency filters

Require: Source Data $\mathscr{D}_{\mathscr{S}} \equiv \{(q_s^i, p_s^i)\}_{i=1}^{m}$, Target unlabeled data $\mathscr{P}_{\mathscr{U}} \equiv \{p_u^i\}_{i=1}^{m^p}$, $\mathscr{Q}_{\mathscr{U}} \equiv \{q_u^i\}_{i=1}^{m_q}$
Ensure: Target domain QG model θ_G, IR model θ_R

Init: $\theta_G, \theta_R \leftarrow$ Train on $\mathscr{D}_{\mathscr{S}}$
repeat
  $S_G \leftarrow [\,], S_R \leftarrow [\,]$ ▷ Synthetic data for θ_G and θ_R
  for $q_u \in \mathscr{Q}_{\mathscr{U}}$ **do**
    $\hat{p} \leftarrow$ Retrieve p from $\mathscr{P}_{\mathscr{U}}$ closest to q_u using θ_R
    add $(\hat{p}, q_u)$ to S_R S_G
  end for
  for $p_u \in \mathscr{P}_{\mathscr{U}}$ **do**
    $\hat{q} \leftarrow$ Generate q from p_u using θ_G
    add $(p_u, \hat{q})$ to S_G S_R
  end for
  $\theta_G \leftarrow$ Finetune on S_G, $\theta_R \leftarrow$ Finetune on S_R
until *dev performance decreases*

3.4.4 Consistency filters for Self-Training and Back-Training

The above algorithms utilize *full* unlabeled data along with their predictions even if the predictions are of low confidence. To alleviate this problem, in self-training, it is common to filter low-confidence predictions [55]. We generalize this notion as *consistency filtering*: For the tasks QG and IR, a *generator* $G \in \{\theta_G, \theta_R\}$ produces synthetic training data for a task whereas the *critic* $C \in \{\theta_G, \theta_R\}$ filters low confidence predictions. We define two types of consistency filtering: 1) **Self consistency** where the generator and critic are the *same*. This is equivalent to filtering out model's own low confidence predictions in self-training. 2) **Cross consistency** where the generator and critic are *different*. This means θ_R will filter the synthetic data generated by θ_G, and vice-versa. For θ_G as critic we use conditional log-likelihood $\log Pr(q|p;\theta_G)$ as the confidence score. For θ_R as critic we use the dot product similarity between the encodings $E_P(p)$ and $E_Q(q)$ as the confidence score. Self-training and back-training can be combined with one or both of the these consistency

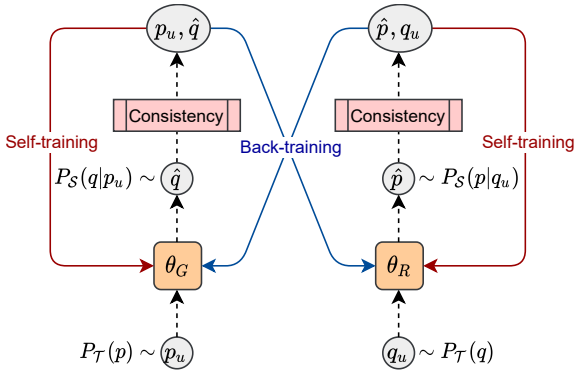

Figure 3.2: Self-training and Back-training for UDA.

checks. We set filter thresholds to accept 75% of synthetic data (refer to section 3.A.1 for exact threshold values).

A popular data filtering technique in data augmentation is cycle consistency [1] which is enforced by further generating noisy input from noisy output, and matching noisy input similarity with source input. We leave its exploration as future work.

3.5 Domain Adaptation Evaluation

As described in Section 3.2, our source domain is NaturalQuestions and the target domains are MLQuestions and PubMedQA. We evaluate if domain adaptation helps to improve the performance compared to no adaptation. We empirically investigate qualitative differences between self-training and back-training to validate their effectiveness. We also investigate if consistency filters and iterative refinement result in further improvements.

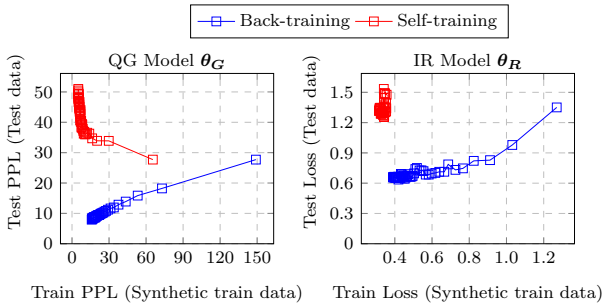

Figure 3.3: Evolution of QG model perplexity (PPL) and IR model loss for Self-training vs Back-training as training proceeds on MLQuestions. *Trajectories run from right to left as training loss decreases with time*. Rightmost points are plotted after first mini-batch training, and subsequent points are plotted after each mini-batch training.

3.5.1 No-adaptation vs. self-training vs back-training

In Table 3.4, we compare the performance of vanilla self-training and back-training (i.e., without consistency filtering or iterative refinement) with the no-adaptation baseline (i.e. model trained on source domain and directly tested on target domain). On MLQuestions, self-training achieves an absolute gain of around 0.6 BLEU-4 points for QG and 7.13 R@20 points for IR. Whereas back-training vastly outperforms self-training, with improvements of 9.4 BLEU-4 points on QG and 19.6 R@20 points on IR over the no-adaptation baseline. The improvements are even bigger on PubMedQA whereas self-training shows no improvement at all.

3.5.2 Why does back-training work?

Figure 3.3 shows the QG model perplexity and IR model loss on synthetic training data and test data as the training (domain adaptation) proceeds on MLQuestions. The plots reveal three interesting

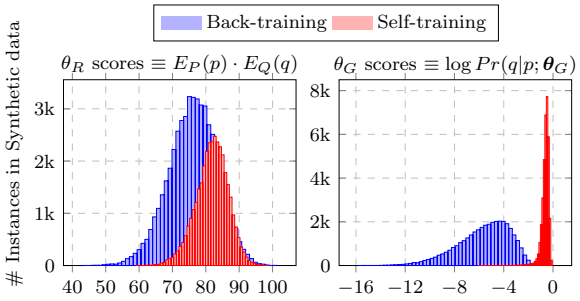

Figure 3.4: DPR embedding similarity scores and QG Log-likelihood scores distribution on MLQuestions synthetic data computed using θ_R and θ_G respectively.

observations: (1) for back-training, the train and test loss (and hence likelihood) are correlated and hence the data generated by back-training matches the target distribution more closely than self-training; (2) self-training achieves lower training error but higher test error compared to back-training, indicating overfitting; (3) extrapolating back-training curve suggests that scaling additional unlabeled data will likely improve the model.

Figure 3.4 plots the distribution for self-training (computing likelihood scores of model's own predictions) and back-training (computing likelihood scores of different model's predictions) for QG and IR tasks on MLQuestions. The figures reveal that self-training curve has *high mean* and *low variance*, indicating less diverse training data. On the other hand, back-training curve has *low mean* and *high variance* indicating diverse training data.

3.5.3 Are consistency filters useful?

Table 3.5 reveals that although our consistency filters outperform base models on MLQuestions, the improvements are not very significant. Our hypothesis is that quality of synthetic data is already

	QG		IR	
Consistency	BLEU4	ROUGE	R@20	R@100
	Self-Training			
None	9.43	32.18	65.26	83.06
Self	**9.85**	**32.34**	64.75	**83.23**
Cross	8.92	31.97	**65.46**	83.00
	Back-Training			
None	18.48	43.97	77.73	91.73
Self	18.62	**44.19**	77.40	91.66
Cross	18.67	43.22	**78.86**	**92.13**

Table 3.5: Effect of using consistency filters on Self-Training and Back-Training for MLQuestions.

high (as backed up by Section 3.5.2 findings), which limits the performance gain. However, the filters reduce synthetic training data by 25%, which leads to faster model training without any drop in performance. Additionally, self-consistency improves self-training in many problems [40, 55]. We believe our cross-consistency filter could also be explored on similar problems.

3.5.4 Is iterative refinement useful?

Further performance improvement of up to 1.53 BLEU-4 points and 2.07 R@20 points can be observed in back-training (Table 3.6) via the iterative procedure described in Algorithm 1. On the other hand, self-training does not show any improvements for QG and marginal improvements for IR.

	QG		IR	
Iteration	BLEU4	ROUGE	R@20	R@100
		Self-Training		
$T = 1$	9.43	32.18	65.26	83.06
$T = 2$	9.28↓	32.09↓	65.60↑	83.78↑
Net Gain	0	0	0.34	0.72
		Back-Training		
$T = 1$	18.48	43.97	77.73	91.73
$T = 2$	20.01↑	46.02↑	79.80↑	93.26↑
Net Gain	1.53	2.05	2.07	1.53

Table 3.6: Evolution of model performance on MLQuestions with increasing iterations: Blue numbers denote increases in performance, while Red numbers denote decrease in performance.

3.5.5 Human Evaluation Results

We also report human evaluation of QG by sampling 50 generated questions from MLQuestions test set and asking three domain experts to rate a question as good or bad based on four attributes: *Naturalness*, i.e., fluency and grammatical correctness; *Coverage*, i.e., whether question covers the whole passage or only part of the passage; *Factual Correctness* in ML domain; *Answerability*, i.e., if the question can be answered using the passage. From the results in Table 3.7, we observe that the back-training model is superior on all four criteria. However, all models perform similarly on *naturalness*.

In Table 3.8 we present some generated questions of various models on MLQuestions and PubMedQA dataset. Subjectively, we find that no-adaptation and self-training models fail to understand domain knowledge, generate generic questions and miss important words present in gold question. Whereas back-training generated question matches more closely to gold question.

Model	N	C	FC	A
No-adaptation	0.64	0.30	0.58	0.68
Self-Training	0.63	0.32	0.58	0.70
Back-Training	**0.66**	**0.41**	**0.64**	**0.88**

Table 3.7: Human evaluations scores between 0-1 on 50 model generated questions for four criteria: Naturalness (N), Coverage (C), Factual Correctness (FC), and Answerability (A).

Passage	Questions
If the line is a good fit for the data then the residual plot will be random. However, if the line is a bad fit for the data then the plot of residuals will be random.	**No-adaptation:** What is the meaning of random plot in statistics? **ST:** What is the meaning of random plot in statistics? **BT:** How do you know if a residual plot is random? **Reference:** How do you know if a residual plot is good?
Financial incentives for smoking cessation in pregnancy are highly cost-effective, with an incremental cost per quality adjusted life years of £482, which is well below recommended decision thresholds.	**No-adaptation:** When do we stop smoking in pregnancy? **ST:** When do you stop smoking in pregnancy? **BT:** Is there a financial incentive for smoking cessation in pregrancy? **Reference:** Are financial incentives cost-effective to support smoking and cessation during pregnancy?

Table 3.8: Examples of generated questions from MLQuestions (first row) and PubMedQA (second row). ST and BT refer to Self-training and Back-training models respectively.

3.5.6 Analysis of Question Types

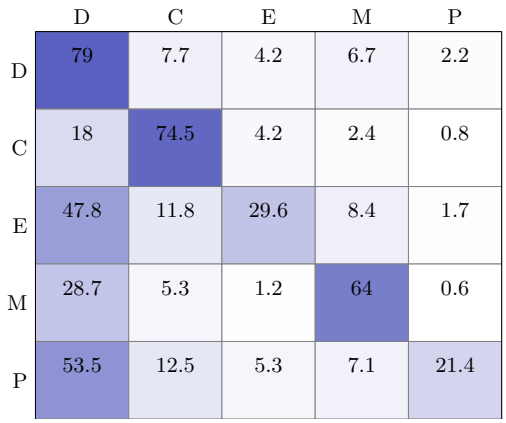

Figure 3.5: Confusion matrix of actual (row) vs model generated question (column) classes for 100 questions sampled from MLQuestions test set. Classes are abbreviated as Description (D) Comparison (C), Explanation (E), Method (M), and Preference (P). Values are in % where each row sums to 100%.

We analyze how well our QG model can generate different kinds of questions according to the taxonomy described in Table 3.2. In Figure 3.5 we plot the confusion matrix between the actual question class and generated question class for our back-training model. To do this, 100 actual questions and corresponding generated questions are sampled from the MLQuestions test set and annotated by a domain expert. We find that the model generates few *Explanation* questions and even fewer *Preference* questions while over-generating *Description* questions. *Comparison* and *Method* questions show good F1-score overall, hence these classes benefit the most from domain adaptation.

3.6 Related Work

Question Generation methods have focused on training neural Seq2Seq models [8, 11, 22, 30, 54] on supervised QA datasets such as SQuAD [35]. Many recent works such as [43, 47] recognize the duality between QG and QA and propose joint training for the two. Duan et al. [12] generate QA pairs from YahooAnswers, and improve QA by adding a question-consistency loss in addition to QA loss. Our work instead establishes strong duality between QG and IR task. Ours is also the first work towards unsupervised domain adaptation for QG to the best of our knowledge.

Passage Retrieval has previously been performed using classical Lucene-BM25 systems [38] based on sparse vector representations of question and passage, and matching keywords efficiently using TF-IDF. Recently, Karpukhin et al. [20] show that fine-tuning dense representations of questions and passages on BERT outperforms classical methods by a strong margin. We adopt the same model for domain adaptation of IR. Concurrent to our work, Reddy et al. [37] also perform domain adaptation for IR. Our focus has been on systematically approaching UDA problem for both QG and IR.

Data Augmentation methods like self-training have been applied in numerous NLP problems such as question answering [9], machine translation [44], and sentiment analysis [17]. Sachan and Xing [40] apply self-training to generate synthetic data for question generation and question answering (QA) in the same domain, and filter data using QA model confidence on answer generated by question.

Back-translation's idea of aligning real outputs with noisy inputs is shared with back-training and has been successful in improving Unsupervised NMT [2, 13]. Zhang et al. [52] use back-translation to generate synthetic data for the task of automatic style transfer. Back-training also shares similarities with co-training [5, 46] and tri-training [25, 48] where multiple models of *same* task generate synthetic data for each other.

3.7 Conclusion and Future Work

We introduce back-training as an unsupervised domain adaptation method focusing on Question Generation and Passage Retrieval. Our algorithm generates synthetic data pairing high-quality outputs with noisy inputs in contrast to self-training producing noisy outputs aligned with quality inputs. We find that back-training outperforms self-training by a large margin on our newly released dataset MLQuestions and PubMedQA.

One area of future research will be exploring back-training for other paired tasks like visual question generation [31] and image retrieval [10], and style transfer [16] from source to target domain and vice-versa. The theoretical foundations for the superior performance of back-training have to be explored further.

3.8 Acknowledgments

We thank the members of SR's research group for their constant feedback during the course of work. We thank Ekaterina Kochmar, Ariella Smofsky and Shayan from Korbit ML team for their helpful comments. SR is supported by the Facebook CIFAR AI Chair and the NSERC Discovery Grant. DK is supported by the MITACS fellowship. We would like to thank SerpAPI for providing search credits for data collection in this work.

Bibliography

[1] Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. Synthetic QA corpora generation with roundtrip consistency. In *ACL*, 2019.

[2] Mikel Artetxe, Gorka Labaka, and Eneko Agirre. Unsupervised statistical machine translation. In *EMNLP*, 2018.

[3] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, 2005.

[4] Yoshua Bengio. Deep learning of representations for unsupervised and transfer learning. In *ICML workshop on unsupervised and transfer learning*, 2012.

[5] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *CoNLL*, 1998.

[6] Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. In *EMNLP*, 2015.

[7] Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *SemEval*, 2017.

[8] Ying-Hong Chan and Yao-Chung Fan. Bert for question generation. In *INLG*, 2019.

[9] Yu-An Chung, Hung-Yi Lee, and James Glass. Supervised and unsupervised transfer learning for question answering. In *NAACL*, 2018.

[10] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys (Csur)*, 2008.

[11] Xinya Du, Junru Shao, and Claire Cardie. Learning to ask: Neural question generation for reading comprehension. In *ACL*, 2017.

[12] Nan Duan, Duyu Tang, Peng Chen, and Ming Zhou. Question generation for question answering. In *EMNLP*, 2017.

[13] Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. Understanding back-translation at scale. In *EMNLP*, 2018.

[14] Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation. In *ACL*, 2018.

[15] Juan M Fernández-Luna, Juan F Huete, Andrew MacFarlane, and Efthimis N Efthimiadis. Teaching and learning in information retrieval. *Information Retrieval*, 2009.

[16] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.

[17] Yulan He and Deyu Zhou. Self-training from labeled features for sentiment analysis. *Information Processing & Management*, 47(4), 2011.

[18] Michael Heilman and Noah A Smith. Good question! statistical ranking for question generation. In *NAACL*, 2010.

[19] Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. Pubmedqa: A dataset for biomedical research question answering. In *EMNLP*, 2019.

[20] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *EMNLP*, 2020.

[21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[22] Tassilo Klein and Moin Nabi. Learning to answer by learning to ask: Getting the best of gpt-2 and bert worlds. *arXiv preprint arXiv:1911.02365*, 2019.

[23] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *TACL*, 2019.

[24] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*, 2020.

[25] Zhenghua Li, Min Zhang, and Wenliang Chen. Ambiguity-aware ensemble training for semi-supervised dependency parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 457–467, Baltimore, Maryland, June 2014. Association for Computational Linguistics. doi: 10.3115/v1/P14-1043. URL https://www.aclweb.org/anthology/P14-1043.

[26] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, 2004.

[27] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[28] Mary L McHugh. Interrater reliability: the kappa statistic. *Biochemia medica*, 2012.

[29] Robert J Menner. The conflict of homonyms in english. *Language*, 1936.

[30] Shlok Kumar Mishra, Pranav Goel, Abhishek Sharma, Abhyuday Jagannatha, David Jacobs, and Hal Daume. Towards automatic generation of questions from long answers. *arXiv preprint arXiv:2004.05109*, 2020.

[31] Nasrin Mostafazadeh, Ishan Misra, Jacob Devlin, Margaret Mitchell, Xiaodong He, and Lucy Vanderwende. Generating natural questions about an image. In *ACL*, 2016.

[32] R Nielsen. Question generation: Proposed challenge tasks and their evaluation. In *Workshop on the Question Generation Shared Task and Evaluation Challenge, Arlington, Virginia*, 2008.

[33] Rodney D Nielsen, Jason Buckingham, Gary Knoll, Ben Marsh, and Leysia Palen. A taxonomy of questions for question generation. In *Workshop on the Question Generation Shared Task and Evaluation Challenge*, 2008.

[34] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002.

[35] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In *EMNLP*, 2016.

[36] Alan Ramponi and Barbara Plank. Neural unsupervised domain adaptation in NLP—A survey. In *COLING*, 2020.

[37] Revanth Gangi Reddy, Bhavani Iyer, Md Arafat Sultan, Rong Zhang, Avi Sil, Vittorio Castelli, Radu Florian, and Salim Roukos. End-to-end qa on covid-19: Domain adaptation with synthetic training. *arXiv preprint arXiv:2012.01414*, 2020.

[38] Stephen Robertson and Hugo Zaragoza. *The probabilistic relevance framework: BM25 and beyond*. Now Publishers Inc, 2009.

[39] Sebastian Ruder, Matthew E Peters, Swabha Swayamdipta, and Thomas Wolf. Transfer learning in natural language processing. In *NAACL: Tutorials*, 2019.

[40] Mrinmaya Sachan and Eric Xing. Self-training for jointly learning to ask and answer questions. In *NAACL*, 2018.

[41] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[42] Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. In *ACL*, 2016.

[43] Duyu Tang, Nan Duan, Tao Qin, Zhao Yan, and Ming Zhou. Question answering and question generation as dual tasks. *arXiv preprint arXiv:1706.02027*, 2017.

[44] Nicola Ueffing. Self-training for machine translation. In *NIPS workshop on Machine Learning for Multilingual Information Access*, 2006.

[45] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *ICML*, 2008.

[46] Xiaojun Wan. Co-training for cross-lingual sentiment classification. In *ACL*, 2009.

[47] Tong Wang, Xingdi Yuan, and Adam Trischler. A joint model for question answering and question generation. *arXiv preprint arXiv:1706.01450*, 2017.

[48] David Weiss, Chris Alberti, Michael Collins, and Slav Petrov. Structured training for neural network transition-based parsing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 323–333, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1032. URL `https://www.aclweb.org/anthology/P15-1032`.

[49] Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *NAACL*, 2018.

[50] David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *ACL*, 1995.

[51] Xiang Yu, Ngoc Thang Vu, and Jonas Kuhn. Ensemble self-training for low-resource languages: grapheme-to-phoneme conversion and morphological inflection. In *SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, 2020.

[52] Zhirui Zhang, Shuo Ren, Shujie Liu, Jianyong Wang, Peng Chen, Mu Li, Ming Zhou, and Enhong Chen. Style transfer as unsupervised machine translation. *arXiv preprint arXiv:1808.07894*, 2018.

[53] Han Zhao, Remi Tachet Des Combes, Kun Zhang, and Geoffrey Gordon. On learning invariant representations for domain adaptation. In *ICML*, 2019.

[54] Yao Zhao, Xiaochuan Ni, Yuanyuan Ding, and Qifa Ke. Paragraph-level neural question generation with maxout pointer and gated self-attention networks. In *EMNLP*, 2018.

[55] Xiaojin Zhu. Semi-supervised learning literature survey. Technical report, Computer Sciences, University of Wisconsin-Madison, 2005.

Authors' addresses

Devang Kulshreshtha School of Computer Science, Mila/McGill University, devang.kulshreshtha@mila.quebec.

Robert Belfer Korbit Technologies Montreal, robert@korbit.ai.

Iulain Vlad Serban Korbit Technologies Montreal, iulian@korbit.ai.

Siva Reddy School of Computer Science, Mila/McGill University, siva.reddy@mila.quebec.

Appendix

3.A Appendix

3.A.1 Model Training Details

All experiments are run with same training configuration. Mean scores across 5 individual runs are provided on the test set. We describe the full model training details below for reproducibility.

BART Question Generation Transformer

We train BART-Base[2] with batch size 32 and learning rate of 1e-5. For all experiments we train the model for 5 epochs, though the model converges in 2-3 epochs. For optimization we use Adam [21] with $\beta_1 = 0.9, \beta_2 = 0.999, \varepsilon = 1e-8$. The question and passage length is padded to 150 and 512 tokens respectively. For decoding we use top-k sampling [14] with $k = 50$. The model is trained with standard cross-entropy objective.

[2]We use huggingface BART implementation `https://huggingface.co/transformers/model_doc/bart.html`

Dense Passage Retriever (DPR)

We use publicly available implementation of DPR model[3] to train our IR system. We also use pre-trained NQ DPR checkpoint provided by authors [4] as the model trained on source domain of NaturalQuestions dataset. The model is trained for 5 epochs with batch size of 32 for all experiments with default hyperparameter settings in Karpukhin et al. [20]. Karpukhin et al. [20] also construct negative examples for each (passage, question) pair where the model maximizes question similarity with gold passage and minimizes similarity with negative passages simultaneously. We construct negative passages similar to Karpukhin et al. [20] as the top-k passages returned by BM25 which match most question tokens but don't contain the answer. We set $k = 7$ for our experiments. For iterative refinement models, we always use same negative passages as the model obtained after 1st iteration ($T = 1$). This is because after each iteration model is being *fine-tuned* starting from previous model and not *re-trained* on pseudo-data. We obtain better performance gains on dev set following this setting.

Consistency Filters

Table 3.A.1 enlists threshold values for different consistency filters. Values are arrived at by plotting confidence scores distribution of synthetic data, and setting threshold to accept 75% of the data (i.e. third quartile Q3). As explained in section 3.4.4, for θ_G as critic we use conditional

[3] https://github.com/facebookresearch/DPR

[4] https://github.com/facebookresearch/DPR/blob/master/dpr/data/download_data.py

Consistency	Critic θ_G	Critic θ_R
Self consistency	-1.19	78.24
Cross consistency	-5.95	71.65

Table 3.A.1: Threshold values for different consistency filters. Values are chosen as the third quartile (Q3) of score distribution of synthetic data, accepting 75% of synthetic data for model training.

log-likelihood $\log Pr(q|p;\theta_G)$ as our confidence scores. For θ_R as critic we use DPR similarity score $E_P(p)E_Q(q)$ as our confidence scores.

3.A.2 NaturalQuestions Dataset Pre-processing

We use Google NaturalQuestions dataset as our *source* domain corpus. We pre-process publicly available train and dev corpora in a similar manner to [30] by selecting all questions starting from the *long-answer* tag and filtering out cases where the long-answer doesn't start with the HTML <p> tag. We obtain 108,501 examples which we split into a 90/10 ratio for training/dev sets. The NQ dev set of 2,136 examples is used as our test data (as the test set is hidden).

3.A.3 MLQuestions: Filtering undesirable data

This section describes filtering out-of-domain questions (OOD) from collected 104K questions from Google described in section 3.2.2. Many ML terms are *homonyms* [29]: they have a different meaning in another context - (e.g. "Ensemble", "Eager Learning", "Transformers"). This means the

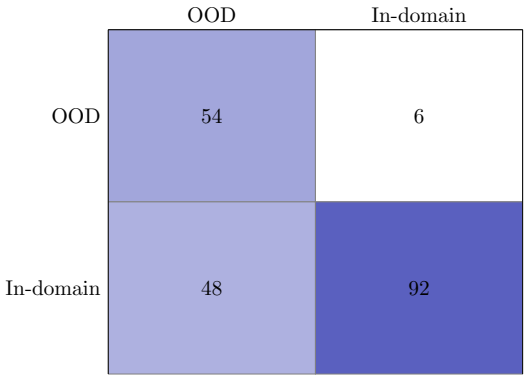

Figure 3.A.1: Test set Confusion matrix of Out-of-domain (OOD) and In-domain classes for classifier probability threshold of 0.8.

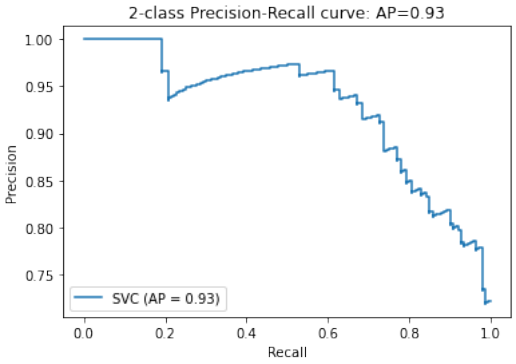

Figure 3.A.2: Precision-Recall curve for Test set of 150 questions. AP denotes average precision.

collected data contains OOD questions. Upon analyzing 100 random questions drawn from 104K questions, we find 27 of them are OOD.

To filter such undesirable data, we randomly sample 1000 questions and recruit 3 domain experts to label them as In-domain or OOD. 200 questions were labeled by all 3 to determine inter-annotator agreement. We record a Cohen's Kappa agreement score [28] of 0.84. The 1000 annotated questions are split into sizes 800, 50, 150 for train, dev, and test sets respectively. Based on this labeled data, we train a classifier on top of question features to classify remaining questions as *useful* or *OOD*. For extracting features from questions, we utilize DistillBERT model [41] trained on SNLI+MultiNLI [6, 49] and then fine-tuned on the STS benchmark[7] train set[5]. This gives us feature vector of size 768 which is used to train SVM classifier[6] with L2 penalty of 0.1. We carefully set the acceptance threshold relatively high to 0.8, to ensure high precision, thus accepting very few OOD questions.

Figure 3.A.1 shows confusion matrix on test set with α set as 0.8. The classifier obtains high precision and average recall of 94.6% and 66% respectively. High precision is empirically verified by annotating 100 random accepted questions, out of which 92 are found to be in-domain. The remaining 8% of the data can be treated as noise for model training. Figure 3.A.2 plots the precision-recall trade-off by varying the acceptance threshold α.

[5]We use off-the-shelf implementation `https://github.com/UKPLab/sentence-transformers` to extract sentence features from pretrained model

[6]`https://scikit-learn.org/stable/modules/svm.html`

3.A.4 Taxonomy of MLQuestions Dataset

In Table 3.2, we show the distribution of various types of questions in MLQuestions and NaturalQuestions dataset. We split the questions into 5 categories based on Nielsen's Educational Taxonomy [33]: *descriptive* questions, which ask for definitions or examples; *method* questions which ask for computations or procedures; *explanation* questions, which ask for justifications; *comparison* questions, which ask to compare two or more concepts; and *preference* questions, which are answered by a selection from a set of options. Refer to Nielsen [32] for detailed understanding of the taxonomy.

3.B Reproducibility Checklist

3.B.1 For all reported experimental results

- *A clear description of the mathematical setting, algorithm, and/or model:* This is provided in Section 3.2 and Section 3.A.1 of the main paper.

- *Submission of a zip file containing source code, with specification of all dependencies, including external libraries, or a link to such resources (while still anonymized):* We provide the source code zipped repository *MLQuestions*. The README file contains all instructions needed to replicate experiments. The file *requirements.txt* specifies required python dependencies.

- *Description of computing infrastructure used:* We perform our experiments on a machine with specifications: 2 CPUs, 2 RTX8000 GPUs, 18GB RAM.

- *The average runtime for each model or algorithm (e.g., training, inference, etc.), or estimated energy cost:* Table 3.B.1 lists average runtime for each step of vanilla Self-training and back-training algorithms, as well as for consistency filters.

- *Number of parameters in each model:* For question generation, the BART base model contains total 139M parameters. For passage retrieval, the DPR model contains total 220M parameters.

- *Corresponding validation performance for each reported test result:* Tables 3.B.2, 3.B.3, 3.B.4 report the validation set performance for each reported test result in the main paper.

- *Explanation of evaluation metrics used, with links to code:* Refer to Section 3.2.6 of main paper for explanation of evaluation metrics. For evaluating QG model, we use the Maluuba NLG-Eval github library to compute BLEU, METEOR, ROUGE scores. The repository can be found at `https://github.com/Maluuba/nlg-eval`. For IR, we implement the top-K retrieval accuracy which can be found in the file location `file://MLQuestions/IR/eval_retriever.py` of our submitted source code.

Task	Data Size	Runtime
θ_G generates synthetic data	50K	174
θ_R generates synthetic data	35K	110
θ_G filters low-quality data	35K	31
θ_G filters low-quality data	50K	45
θ_G filters low-quality data	35K	35
θ_G filters low-quality data	50K	48
θ_G Self-training	50K	373
θ_G Back-training	35K	263
θ_R Self-training	35K	547
θ_R Back-training	50K	762

Table 3.B.1: Runtime (in minutes) for each step in domain adaptation models for MLQuestions dataset. Since there are 35K unaligned questions and 50K unaligned passages, a step has different execution times depending on type of training (self/back) or consistency filter (self/cross).

3.B.2 For all experiments with hyperparameter search

- *The exact number of training and evaluation runs:* For all experiments we train the QG and IR for 5 epochs. We evaluate the model performance using evaluation metrics after each epoch on the validation set, and find that models converge after 2-3 epochs.

- *Bounds for each hyperparameter:* We experimented by manually varying hyperparameters in vicinity of values mentioned in section 3.A.1. The best hyperparameters on validation set were chosen for final model training.

- *Hyperparameter configurations for best-performing models:* We provide complete hyperparameters details for QG and IR model in Section 3.A.1.

- *The method of choosing hyperparameter values (e.g., uniform sampling, manual tuning, etc.)*

Dataset	Model	Question Generation						Passage Retrieval			
		B1	B2	B3	B4	M	R	R@1	R@20	R@40	R@100
MLQuestions	No-adaptation	30.64	19.70	12.82	8.80	23.23	31.33	13.00	54.86	64.6	73.93
	Self-Training	31.01	20.36	13.50	9.37	23.67	31.75	14.13	62.20	70.80	80.66
	Back-Training	**41.42**	**30.72**	**22.50**	**17.29**	**23.38**	**41.58**	**21.86**	**77.40**	**84.66**	**90.26**
PubMedQA	No-adaptation	14.23	7.02	3.65	1.81	9.12	15.96	32.66	57.20	61.8	72.68
	Self-Training	14.04	6.98	3.09	1.54	8.67	15.30	33.0	57.48	64.2	73.44
	Back-Training	**27.17**	**17.92**	**12.34**	**8.76**	**17.66**	**25.89**	**56.60**	**81.0**	**83.20**	**87.68**

Table 3.B.2: Validation set results of unsupervised domain adaptation. *No-adaptation* denotes the model trained on NaturalQuestions and evaluated directly on MLQuestions/PubMedQA dev sets without any domain adaptation.

	QG		IR	
Consistency	BLEU4	ROUGE	R@20	R@100
	Self-Training			
None	9.37	31.75	62.20	80.66
Self	**9.76**	31.67	62.75	81.56
Cross	9.02	**32.34**	**62.96**	**82.00**
	Back-Training			
None	17.29	41.58	77.40	90.26
Self	17.91	**43.27**	76.86	91.06
Cross	**18.09**	41.84	**78.26**	**91.33**

Table 3.B.3: Effect of using consistency filters on Self-Training and Back-Training for MLQuestions validation set.

and the criterion used to select among them (e.g., accuracy): We use manual tuning method with the criterion as BLEU-4 accuracy for QG and R@40 retrieval accuracy for IR task on validation set.

- *Summary statistics of the results (e.g., mean, variance, error bars, etc.):* Mean scores across 5 individual runs are provided for all experiments of main paper.

	QG		IR	
Iteration	BLEU4	ROUGE	R@20	R@100
		Self-Training		
$T = 1$	9.37	31.75	62.20	80.66
$T = 2$	9.22↓	31.13↓	62.80↑	81.08↑
Net Gain	0	0	0.60	0.42
		Back-Training		
$T = 1$	17.29	41.58	77.40	90.26
$T = 2$	19.97↑	45.74↑	78.56↑	91.26↑
Net Gain	2.68	4.16	1.16	1.00

Table 3.B.4: Evolution of model performance on MLQuestions validation set with increasing iterations: Blue numbers denote increases in performance, while Red numbers denote decrease in performance.

3.B.3 For all datasets used

- *Relevant details such as languages, and number of examples and label distributions:* Section 3.2 provide statistics of NaturalQuestions, MLQuestions, and PubMedQA datasets. All datasets are in English language.

- *Details of train/validation/test splits:* This is also provided in section 3.2 for all three datasets.

- *Explanation of any data that were excluded, and all pre-processing steps:* Relevant details are provided in section 3.2 for all three datasets.

- *A zip file containing data or link to a downloadable version of the data:* We provide MLQuestions dataset in the submission zip file. The NaturalQuestions and PubMedQA dataset can be downloaded from `https://ai.google.com/research/NaturalQuestions/download` and

`https://github.com/pubmedqa/pubmedqa` repsectively. The datasets can be pre-processed following the procedures mentioned in section 3.2.

- *For new data collected, a complete description of the data collection process, such as instructions to annotators and methods for quality control.:* We provide above details for our newly created dataset *MLQuestions* in section 3.2.2.

Chapter 4

Paper II: Few-shot Question Generation for Personalized Feedback in Intelligent Tutoring Systems

Devang Kulshreshtha, Muhammad Shayan, Robert Belfer, Siva Reddy, Iulian Vlad Serban, Ekaterina Kochmar

Published in *Proceedings of the 11th International Conference on Prestigious Applications of Intelligent Systems, PAIS 2022*.

Problem Statement: Showing corrective feedback to students while solving Q/A exercises in educational ITS is helpful to improve their learning gains. The automated feedback should be highly domain-aware and personalized which can pinpoint correct/incorrect phrases in student answers and guide them towards correct answer. In the following chapter, I explore automatically generated questions as personalized feedback in an ITS, which follows above mentioned requirements and outperforms competitive baselines in student learning gains by a significant amount.

Abstract

Existing work on generating hints in Intelligent Tutoring Systems (ITS) focuses mostly on manual and non-personalized feedback. In this work, we explore automatically generated questions as personalized feedback in an ITS. Our personalized feedback can pinpoint correct and incorrect or missing phrases in student answers as well as guide them towards correct answer by asking a question in natural language. Our approach combines cause–effect analysis to break down student answers using text similarity-based NLP Transformer models to identify correct and incorrect or missing parts. We train a few-shot Neural Question Generation and Question Re-ranking models to show questions addressing components missing in the student's answers which steers students towards the correct answer. Our model vastly outperforms both simple and strong baselines in terms of student learning gains by 45% and 23% respectively when tested in a real dialogue-based ITS. Finally, we show that our personalized corrective feedback system has the potential to improve Generative Question Answering systems.

Exercise Problem: We want to choose between 2 treatments A and B. For both, we got same mean recovery rate but higher variance for treatment A. Which treatment would you discard, and why?	
Student: Treatment A	*Student:* Treatment A
System [Non-personalized]: That's not right. Look at the variances and provide an explanation why you think one treatment is better than the other.	*System [Personalized]:* "Treatment A" is correct! Try supplying a reason for this idea. Do we prefer more homogeneous results or less? *Student:* Less *System:* Ok, now try to answer original exercise.
Student: Treatment B?	*Student:* Treatment A, because it is less homogeneous than treatment B.
System: Not really. Let's move to another problem.	*System:* That's correct!

Table 4.1.1: Non-Personalized vs Personalized Feedback Generation in Korbit ITS. The Personalized Feedback pinpoints correct and missing parts in the answer and provides suggestions on how to improve it. In this case, the student forgot to provide reasoning for their answer and is asked a question about the missing part.

Keywords. Intelligent tutoring systems, Natural language processing, Deep learning, Question Generation, Personalized learning and feedback

4.1 Introduction

Intelligent Tutoring Systems (ITS) are AI-powered instructional systems that provide personalized teaching to students [38]. ITS are a low-cost alternative to conventional classroom teaching, and shown to be more effective for tutoring students [34, 35]. One of the critical aspects of ITS is the ability to provide personalized feedback for exercises.

Many ITS however rely heavily on expert hand-crafted rules to generate feedback which becomes infeasible for large amounts of educational texts. An important research goal is to thus develop automated feedback systems from student-tutor conversation history [24, 26]. Existing

work mainly focuses on non-personalized hints created using template-based methods [2, 22]. However, students make various type of mistakes (such as grammatical errors, correct answers with incorrect reasoning, and so on), and showing the same hint to address different mistakes is not efficient in improving students' answers, and might even further confuse them. As a result, this can lead to lower motivation and a decrease in the overall study time spent on an ITS platform.

In this paper, we propose a novel automated personalized feedback system based on deep-learning based Transformer models [20, 40] to address the above-mentioned problems. Our model first breaks apart student answer into various components by performing cause-effect relation extraction [4]. Then it matches the components with gold answers using similarity-based Transformers [40], and classifies them into various error categories (such as *missing explanation, incorrect main answer*, and so on). Next, a few-shot Transformer [28] model generates a personalized natural language question which is combined with the output of the cause–effect analysis to generate question-based feedback. We integrate the feedback in the conversation between an AI-tutor and a student. Such questions are easier to answer compared to the original exercises, as they are aimed at guiding a student towards improving their response.

Table 4.1.1 demonstrates a real interaction with the feedback system. Consider the case where student supplies the correct answer without an explanation. The non-personalized model produces a generic hint irrespective of any student answer, saying *'Thats not right'* even though the main answer is correct. This further confuses the student and causes them to change their correct answer in the next attempt. In contrast, our personalized model first informs the student that their answer

is correct and prompts them to supply explanation. It then asks a clarifying question steering the student towards the reasoning part which was missing. As a result, the student is able to provide a correct solution.

We test our method on Korbit ITS,[1] a large-scale AI-powered personalized ITS. Students watch video lectures on data science topics and working on problem-solving exercises created by domain experts. While going through exercises, the student's answers are compared to reference solutions using an ML-based solution checker. We trigger hint generation when the Korbit's solution checker model marks a student answer as incorrect. We measure the student learning gains after showing our feedback. Our approach outperforms a minimal feedback (simple) baseline by 45% and personalized human feedback (strong) baseline by 23%.

4.2 Background: Exercises in Korbit ITS

Each exercise in Korbit consists of a problem text, and one or more reference solutions. We focus on a particular class of exercises and name them as *cause-effect* exercises. In these exercises, the student is asked about identifying one or several relevant concepts, but they also require to justify the explanation behind their answer. An example of such exercise is *'Can linear regression be applied to classification? Why or why not?'*. Here the expected solution can be decomposed into an *answer (effect)* and *explanation (cause)*. For example, an acceptable solution to the problem above can be *'No, as the output variable of linear regression is continuous'*. Here, the cause is *'The output*

[1] https://www.korbit.ai/

Connective	Reference solution
because	It's a discrete variable *because* it's counting the number of vehicles
,	No, the feature has 0 weight in the model function.
then	If the output is over the threshold *then* *x* is fraudulent

Table 4.2.1: Decomposition of reference solutions in Korbit ITS into their cause and effect.

variable of linear regression is continuous' and effect is *'No'*. Table 4.2.1 illustrates more such examples.

Cause-effect exercises require critical reasoning to solve, as opposed to reading comprehension exercises such as SQuAD [29]. The explanation component can not be usually found directly in any pre-existing knowledge bases or paragraphs. Due to this, the need for personalized feedback is higher in such exercises.

4.3 Personalized Feedback Generation Model

Our model generates feedback in three steps - (i) error classification (ii) Question Generation (iii) Full feedback generation. They are illustrated in Figure 4.3.1 and detailed below:

4.3.1 Cause-Effect Error Classifier

Decomposing a solution into its cause (explanation) and effect (answer) allows classification of student errors. Denote student solution as $s_s \equiv \{c_s, e_s\}$ and gold solution as $s_r \equiv \{c_r, e_r\}$ decomposed into their cause c and effect e by running a cause-effect extractor described in Cao et al. [4]. The student deficiency falls into one of the four categories:

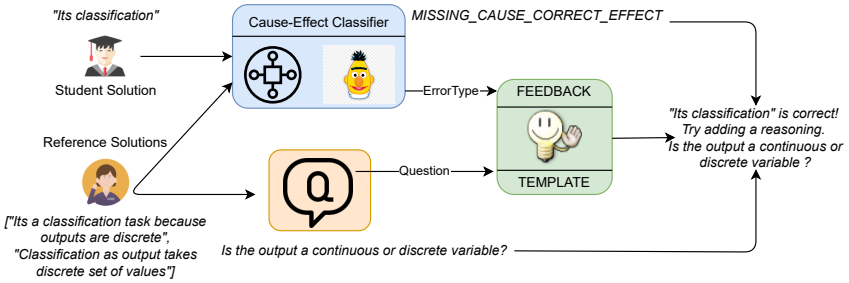

Figure 4.3.1: An overview of our personalized feedback generation system: (a) Student solution is classified into its error type using cause-effect extractor and BERT similarity. (b) A few-shot QG model generates question from the *cause* of reference solution. (c) Personalized hint is generated using different feedback templates.

- Incorrect Cause $[c_s \neq c_r]$ Incorrect Effect $[e_s \neq e_r]$
- Correct Cause $[c_s \equiv c_r]$ Incorrect Effect $[e_s \neq e_r]$
- Incorrect Cause $[c_s \neq c_r]$ Correct Effect $[e_s \equiv e_r]$
- Missing Cause $[c_s \equiv \varnothing]$ Correct Effect $[e_s \equiv e_r]$

Figure 4.3.2 describes examples of all errors for a given exercise, as well as the error distribution generated by running cause-effect extractor over $7,000$ incorrect solutions.

To detect the error type, we match student cause-effect text with reference solution using BERTScore [40]. BERTScore uses pre-trained BERT [6] contextualised embeddings and computes overall similarity using weighted mean of cosine similarity between their tokens. It correlates better with human judgments compared with n-gram overlap based metrics (e.g. BLEU, ROUGE etc). BERTScore has been used as an evaluation metric for image captioning [40], summarization ([21]), machine translation ([37]) etc. BERTScore returns a score $(0-1)$ between student and reference

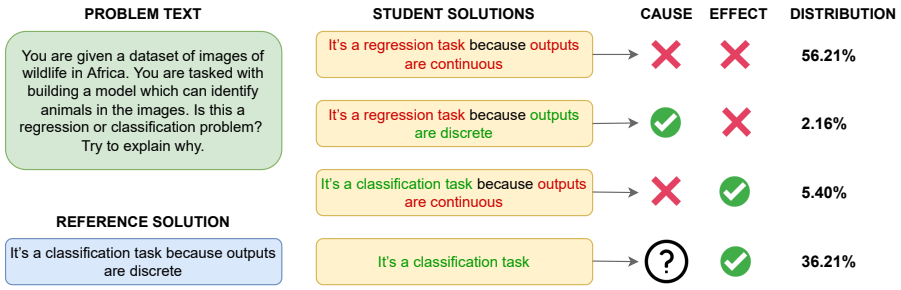

Figure 4.3.2: Illustrating various types of student errors for a *cause-effect* exercise in Korbit ITS.

cause/effect. If similarity exceeds a threshold ($= 0.8$, set manually) then cause/effect is considered correct.

4.3.2 Few-shot Question Generation

Our goal is to generate a question which forces the student to think about the incorrect/missing components in their solutions, and improve their answers. Our QG model pipeline comprises of four steps described below:

Dataset creation

We create a dataset by randomly sampling around 112 cause-effect exercises, giving us around 300 reference solutions for those exercises. We then ask four domain experts to write a question from the reference solutions, giving 75 examples to each annotator. Questions are written to *not* reveal the *effect/answer* and hence created only from *explanation* of reference solution. The annotators mainly write three type of questions - open-ended, binary, and binary with alternatives. Examples of such types are shown in Table 4.3.1 (the 'Score' column will be explained later in Section 4.3.2). All

Reference Solution: *It is classification because coin flip outcome is discrete.*		
Question Type	**Example**	**Score**
Binary	Is flipping a coin discrete?	0.5
Binary with alternatives	Is flipping a coin discrete or continuous?	0.8
Open-Ended	What kind of action is flipping a coin?	1

Table 4.3.1: Taxonomy of questions written by annotators and corresponding scores used for question re-ranking.

annotators also annotate a common set of 20 questions to ensure that annotators have low variance in questions. We find the common questions are quite similar and follow the above guidelines.

Few-Shot Question Generation (QG) model

After data collection, we train a QG model to generate questions from reference solutions. We frame QG as a neural sequence-to-sequence task similar to [8] where an encoder reads input text and decoder produces question by predicting one word at a time. We experiment with two pre-trained Transformers: BART [20] and T5 [28].

T5 is an encoder-decoder model pre-trained on a mixture of supervised and unsupervised NLP tasks where each task is converted into text-to-text input-output. T5 works well on a variety of conditional sequence generation tasks such as summarization [31], machine translation and question generation [7]. We name the model as *T5-QG*.

BART is a Transformer autoencoder pre-trained to reconstruct text from noisy text inputs. For QG, it learns a conditional probablity distribution $P(q|r)$ to generate question q from reference solution r. We experiment with two pre-trained checkpoints - a) original BART-base checkpoint

provided by authors and b) BART model trained on 50K MLQuestions dataset using back-training algorithm [17]. The latter model is able to generate good-quality questions for data science domain which is also our domain of interest in Korbit ITS. We denote them as *BART-QG* and *BART-ML-QG*.

We split the data into 220 train, 40 validation and 40 test examples to train these models. Refer to Section 4.A for model training details.

Improving Question Generation using Re-Ranking

To improve question quality, we train a question re-ranker to chose the best question: First we generate $k = 3$ questions per reference solution using beam search decoding algorithm [10] for 80 randomly sampled reference solutions. Then we ask four domain experts to rate the usefulness of 240 generated questions on a scale of 1-5. Rating is done keeping in mind the *factual correctness, fluency* and *relevance* of question to the input reference solution. Additionally, good quality questions based on question type are given higher score based on the preference - *{open-ended > binary with alternatives > binary}* question (see Table 4.3.1 for question type examples). The 240 examples are distributed equally amongst three annotators. We find that the mean ratings given by each annotator was quite similar - $3.35, 3.4, 3.46$. Additionally a common set of 20 examples are annotated by all annotators and we record an inter-annotator agreement of 0.75 which shows substantial agreement according to Landis and Koch [19].

Finally we train a Linear Regression model to predict usefulness taking the reference solution and generated question as input on 200 examples, and test on 40 examples. The input features to

the regression model are -

- **Sentence Embeddings:** We use Sentence-BERT [30] to extract 768 dimensional embeddings from question. The Sentence-BERT uses siamese and triplet network structures to derive sentence embeddings from BERT and have been shown to perform extremely well in common STS tasks and transfer learning tasks [30].

- **Well-formedness:** We train a BERT binary classifier to predict whether a question is well-formed or ill-formed on Google Well-formedness dataset [9]. We use the well-formedness probability of generated hint question as the *well-formed* feature.

- **Fluency:** We finetune a GPT-2 LM [3] on the 300 original hand-written questions (Section 4.3.2) using causal language modeling (LM) objective. The negative of LM perplexity of generated question is used as *fluency* feature.

- **Model Confidence:** This feature is computed as the negative loss of model when the generated question is considered as ground truth.

- **Question Type:** We want to penalise simple questions and reward questions which are more diverse and challenging to answer. We come up with a simple heuristic depicted in Table 4.3.1 to compute question type score feature of a question.

We get 772 dimensional feature vector and train our regression model using Ordinary Least Squares (OLS) objective on 200 examples. During inference, we use this question re-ranker to select the best question from the 5-best list for each reference solution.

After training the Question Generation and reranker model, we generate questions from all 1470 reference solutions in Korbit ITS using above models.

4.3.3 Providing Feedback

Using the output of cause-effect classifier and question generator, we provide feedback to reveal student deficiencies and suggest improvements. First we find the reference solution s_r closest to student solution s_s using BERTScore similarity. Then according to each error category identified by cause-effect classifier in Section 4.3.1, we create feedback described below (full algorithm is described in Algorithm 2)-

Incorrect Cause [$c_s \neq c_r$] Incorrect Effect [$e_s \neq e_r$] First the system reveals error type by saying - *"$\{e_s\}$ is incorrect."*. Then it asks a question generated from s_r using QG model. Then the student responds to that question, after which we completely ignore their answer. The system then asks the student to answer the original exercise again.

Incorrect Cause [$c_s \neq c_r$] Correct Effect [$e_s \equiv e_r$] Since the main answer (effect) is correct, first the system outputs - *"$\{e_s\}$ is correct! Try changing your reasoning."*. Then similar to previous error type, we ask a sub-question generated from s_r. After student answers this sub-question, the interface will ask them to answer original exercise again.

Algorithm 2 Personalized Feedback Generation in Korbit ITS

Require: Exercise problem Q, reference answers $\mathscr{R} \equiv \{s_r^i\}_{i=1}^m$, incorrect student answer s_s, Cause-Effect Extractor θ_{CE}, BERTScore model θ_{BS}, BERTScore similarity threshold τ_{BS}, Question Generator θ_{QG}.
Ensure: Personalized hint h

/*Find reference answer closest to student answer*/
$sim \leftarrow []$
for $s_r \in \mathscr{R}$ **do**
    add $\theta_{BS}(s_r, s_s)$ to sim
end for
$s_r \leftarrow \arg\max_i(sim)$
/*Classify student error and generate personalized hint*/
$(c_r, e_r) \leftarrow \theta_{CE}(s_r); (c_s, e_s) \leftarrow \theta_{CE}(s_s)$ ▷ Run cause-effect extractor
$q = \theta_{QG}(s_r)$ ▷ Generate question from reference solution.
switch $[c_r, e_r, c_s, e_s]$ **do**
    case $c_s \neq c_r$ **and** $e_s \neq e_r$ ▷ $[\theta_{BS}(c_s, c_r) < \tau_{BS}$ **and** $\theta_{BS}(e_s, e_r) < \tau_{BS}]$
        return *"{e_s} is incorrect. {q}?"*
    case $c_s \neq c_r$ **and** $e_s \equiv e_r$ ▷ $[\theta_{BS}(c_s, c_r) < \tau_{BS}$ **and** $\theta_{BS}(e_s, e_r) \geq \tau_{BS}]$
        if $c_s \equiv \varnothing$ **then**
            return *"{e_s} is correct! Try supplying a reason for it. {q}?"*
        else
            return *"{e_s} is correct! Try changing your reasoning. {q}?"*
        end if
    case $c_s \equiv c_r$ **and** $e_s \neq e_r$ ▷ $[\theta_{BS}(c_s, c_r) \geq \tau_{BS}$ **and** $\theta_{BS}(e_s, e_r) < \tau_{BS}]$
        return *"Did you mean {e_r} because {c_s}?"*

Missing Cause [$c_s \equiv \varnothing$] Correct Effect [$e_s \equiv e_r$] We show similar hint as previous error category, saying - *"{e_s} is correct! Try supplying a reason for it. {q}?"*, where q is the generated question. This example is also illustrated in Figure 4.3.1.

Correct Cause [$c_s \equiv c_r$] Incorrect Effect [$e_s \neq e_r$] In practice this scenario rarely occurs. Since student supplied correct explanation, it is likely that student supplied incorrect answer by mistake. In this case we repair student's solution by asking an MCQ question - *"Did you mean {e_r} because {c_s}?"*. The interface supplies two options to chose from - "Yes, I agree" and "No, I disagree". If the student chooses former option then answer is marked correct, otherwise incorrect.

Model	B1	B2	B3	B4	R
T5-QG	30.4	18.0	11.9	7.5	30.7
BART-QG	34.5	24.5	17.1	12.1	39.3
BART-ML-QG	**36.1**	**24.7**	**19.6**	**12.2**	**39.7**

Table 4.4.1: Results of Question Generation Models on standard language evaluation metrics.

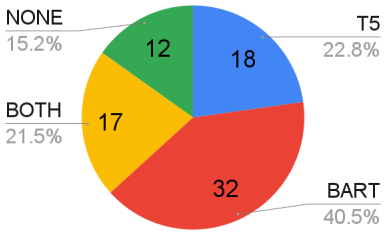

Figure 4.4.1: Comparing question quality of T5 with BART based on annotated 80 questions.

4.4 Experimental Results

4.4.1 Question Generation

We evaluate the generation quality of three models - *T5-QG, BART-QG, BART-ML-QG* using standard language generation metrics: BLEU1-4 [27] and ROUGE-L [32] on the test set of 40 examples. The results are presented in Table 4.4.1. We find BART outperforms T5 by 4 BLEU1 points, showing that BART is better suited for conditional generation. Also pre-training on MLQuestions dataset [17] increases BLEU1 by 1.5 absolute points.

4.4.2 Question Re-ranking

For question re-ranking, we experiment using different combinations of features described in Section 4.3.2 to predict usefulness score of generated question -

1. **Mean Baseline:** This baseline simply outputs the usefulness as the average of all usefulness output in training set.

2. **Linguistic:** Here we only use the four linguistic features - *well-formedness, fluency, model confidence, question type score* as features for the question re-ranker.

3. **SBERT:** Here we only use the 768-dimensional SBERT embedding features.

4. **Ling-SBERT:** In this model we concatenate SBERT sentence embeddings with four linguistic features to train our question re-ranker.

For each model we measure standard regression evaluation metrics - Mean Squared Error (MSE), Mean Absolute Error (MAE), Pearson Correlation (PCR). We also measure usefulness metric for each model. To compute usefulness, the re-ranker model predicts usefulness for each of the $k = 3$ questions of the same given reference solution from the test set. Then the gold (actual) usefulness label for question achieving highest score is averaged across all reference solutions in test set. The results are presented in Table 4.4.2. We find that Ling-SBERT outperforms all other models for all metrics. More importantly, it improves the usefulness rating from 3.42 to 4.01. This means incorporating question re-ranking improves the actual usefulness of question generation by 0.5 on average!

Model	MSE	MAE	PCR	Usefulness
Mean Baseline	2.20	1.32	-	3.42
SBERT	1.74	1.16	0.38	3.96
Linguistic	1.78	1.16	0.33	3.85
Ling-SBERT	**1.72**	**1.15**	**0.40**	**4.01**

Table 4.4.2: Results of Question Re-ranking.

4.4.3 Human Evaluation

We manually compare the question quality generated by *T5-QG* and *BART-ML-QG* by generating questions for 80 randomly sampled reference solutions. The annotators compare both questions and provide one of the four labels - T5 (meaning T5 question is more useful than BART), BART (BART question is more useful), BOTH (both are equally good), and NONE (neither is a good question). The results from Figure 4.4.1 indicate that BART model is the clear winner, which is also supported by the superior BLEU scores.

Based on results on question generation, re-ranking and human evaluation we use the BART-ML-QG model for generation, and the Ling-SBERT model for re-ranking.

4.4.4 Student Learning Gains

After integrating our models in Korbit ITS, we collect around 146 distinct student interactions with feedback system for 550 exercises and measure student learning gains. The student learning gain is defined as the percentage of times a student answer is labelled correctly by the solution checker after they have received a given feedback. We compare our *Personalized Question-based Feedback* with both simple and strong baselines:

- **Minimal Feedback Baseline:** Here the system will simply tell the student that their solution is incorrect and they should try again.

- **Personalized Human Feedback Baseline:** For every exercise, Korbit already has several hints manually crafted by course designers. To select the best hint from the ones available, the ITS uses a personalized ML model by looking at student performance and responses on the exercise [16]. This personalization is used only during hint selection, and *not* during hint generation itself (which is manual).

- **Personalized Non-question Feedback:** In this model after informing error type using cause-effect classifier, we reveal a part of the answer rather than asking a question. For e.g. if the student answers *'Its a regression task because outputs are continuous'*, we show the hint as *'"Its a regression task" is incorrect. Observe that outputs are discrete'* and ask the student to try again.

We present results of student learning gains in Table 4.4.3. The 'First Attempt' column indicates entries in which the student tried only once previously, while 'All Attempts' considers learning gains across student's all attempts. Our experiments show that our *Personalized Question-based Feedback* model outperforms all models.

The *Non-question Feedback* model improves over *minimal feedback baseline* by 18%, because it additionally informs about correct and incorrect/missing components. However, it cannot tell the student how to correct the incorrect/missing part. Our *Question-based Feedback* model further

Model	Average Learning Gain (%)	
	First Attempt	All Attempts
Minimal Feedback Baseline	22.58 ± 14.72	21.74 ± 11.92
Personalized Human Feedback Baseline	31.25 ± 16.06	30.43 ± 13.3
Personalized Non-question Feedback	41.67 ± 19.72	34.38 ± 16.46
Personalized Question-based Feedback	**66.67 ± 16.87**	**52.27 ± 14.76**

Table 4.4.3: Student learning gains on the Korbit ITS at 95% confidence intervals.

improves over it by 26%. This shows that asking questions about missing/incorrect parts is the key to help students improve their answers.

For all models, we observe that learning gains for 'First Attempt' are more than 'All Attempts'. This is likely because students who require many hints to solve an exercise may have knowledge gaps to solve exercises.

We find that most frequent student error is *'incorrect cause incorrect effect'* followed by *'missing cause correct effect'*. The error type *'Correct cause incorrect effect'* occurs rarely as students usually know the main answer if they know the explanation behind it.

4.5 Improving Generative Question Answering using Feedback Intervention

Will a student having access to a feedback generation to correct it's mistakes during training perform better than another student without the feedback system support? Assume Student S_A and S_B are being taught by instructors I_A and I_B. I_A trains S_A by showing the answer for many questions. While

I_B trains S_B by showing answers for questions, as well as sending *personalized corrective feedback* when student answers question incorrectly. During test time, both students get same question paper without access to any feedback.

We simulate this behaviour by replacing student by QA model and teacher by hint model:

1. Train baseline QA model θ_{QA} to generate reference solution from question.
2. Generate machine (student) answers for questions in training data using θ_{QA}. Generate hints using our feedback system described previously for machine generated (incorrect) answer.
3. Train hint generator θ_{HG} to generate these hints from question & machine answer.
4. Train hint-assisted QA model θ_{HQA} to generate answer from question and hint text generated by θ_{HG}.
5. During inference, first generate machine answer using θ_{QA}. Next generate hint using θ_{HG} then generate final answer using θ_{HQA}.

The full algorithm is described in 3. In principle we are first generating intermediate hint (part of answer) and then using it to generate the full answer. Similar inductive bias to learn the output in parts has been show to improve models in QA [18] and QG [13].

4.5.1 Hint-Answer Entailment Consistency

In the above model, it is logical to expect the generated hint and model answer should be consistent with each other i.e. *machine answer should **entail** model hint*. How can we enforce this inductive

Algorithm 3 Improving Generative QA using Personalized Feedback Generation

Require: QA Data $\mathscr{D}_{QA} \equiv \{(q^i, a^i)\}_{i=1}^m$, Personalized Hint Generator $\mathscr{H}$
Ensure: Hint assisted QA model θ_{HQA}

$\theta_{QA} \leftarrow$ Train on $\mathscr{D}_{QA}$ ▷ Vanilla QA model
$\mathscr{D}_{HG} \leftarrow [\,]$ ▷ Synthetic data for θ_{HG}
for $q, a \in \mathscr{D}_{QA}$ **do**
    Generate machine answer $\hat{a} = \theta_{QA}(q)$
    Generate personalized hint $h = \mathscr{H}(q, \hat{a}, a)$
    add $(q, \hat{a}, h)$ to $\mathscr{D}_{HG}$
end for
$\theta_{HG} \leftarrow$ Train on $\mathscr{D}_{HG}$ to generate h from $(q, \hat{a})$
$\mathscr{D}_{HQA} \leftarrow [\,]$ ▷ Synthetic data for θ_{HQA}
for $q, a \in \mathscr{D}_{QA}$ **do**
    Generate machine answer $\hat{a} = \theta_{QA}(q)$
    Generate hint $\hat{h} = \theta_{HG}(q)$
    add $(q, \hat{h}, a)$ to $\mathscr{D}_{HQA}$
end for
$\theta_{HQA} \leftarrow$ Train on $\mathscr{D}_{HQA}$ to generate a from $(q, \hat{h})$

bias in the model? During inference, we generate $k = 3$ model answers and measure the entailment probability of each answer to model generated hint using entailment probability of RoBERTa model [2] trained on multiple entailment datasets [25]. We pick the model answer having the highest entailment probability.

4.5.2 Experiments and Results

We use BART to train θ_{QA}, θ_{HG}, and θ_{HQA}. Refer to 4.A for model training details. Since there exists no generative cause-effect QA dataset to the best of our knowledge, we use Korbit dataset of 550 exercises and reference solutions. We split the data into 400 train, 50 validation and 100 test examples and measure BLEU and ROUGE metrics.

[2] https://huggingface.co/ynie/roberta-large-snli_mnli_fever_anli_R1_R2_R3-nli

Models	BLEU1	BLEU2	BLEU3	BLEU4	ROUGE-L
Vanilla-QA	24.57	14.89	10.70	8.27	29.68
Hint-assisted QA	25.16	15.07	11.56	9.37	30.63
Hint+Entailment	**25.54**	**16.05**	**12.19**	**9.57**	**31.35**

Table 4.5.1: Results on improving Generative Question Answering Using Hint Intervention

Experimental results presented in Table 4.5.1 demonstrate that Hint-assisted QA system is superior to Vanilla-QA model by 1 ROUGE point, and enforcing hint-answer entailment further boosts ROUGE by up to 1.5 points. Although the improvements are marginal, note that the task itself is hard as the training data is limited.

4.6 Related Work

Feedback Generation Previous research on dialogue-based ITS similar to Korbit investigated various aspects of automated feedback generation and adaptation [1, 23, 36]. In particular, Stamper et al. [36] investigated ways to augment their Deep Thought logic tutor with a Hint Factory that generated data-driven, context-specific hints for an ITS. The hints were effective in promoting learning, however, their approach mostly focused on the automated detection of the best hint sequence among hints consisting of logic rules, whereas our work focuses on methods of hint generation in natural language. The most similar work to ours is that Grenander et al. [11], who also generate personalized feedback based on cause–effect analysis, but do not use questions in their generated feedback, hence their feedback does not reveal any hint about correct answer.

Question Generation Previous research has focused on training neural Seq2Seq models [8, 15, 41] on supervised full QA datasets such as SQuAD [29]. QG in a few-shot setting under limited data has also been explored recently for multi-hop QG [12, 39].

Chen et al. [5] create a large-scale Educational QG dataset from KhanAcademy and TED-Ed data sources as a learning and assessment tools for students. Kulshreshtha et al. [17] also release a QG dataset comprising of data-science questions to promote research in domain adaptation. Unlike our questions, the questions in Chen et al. [5], Kulshreshtha et al. [17] are static and not personalized to the student. A recent work by Srivastava and Goodman [33] generates personalized questions according to the student's level by proposing a difficulty-controllable QG model. To the best of our knowledge, we are the first to use QG in an education context with real student interaction data.

Improving Question Answering using Hints is not yet studied clearly in NLP paradigm. A related work by Lamm et al. [18] proposes the use of *explanations* for an answer to improve Question Answering. They annotate a dataset of 8,991 QED explanations and use it to learn joint QA and explanation generation. Their explanations however are very different from our hints as they are non-personalized (fixed for a given question/answer).

4.7 Conclusion and Future Work

We show how can we provide personalized feedback to students in an ITS by combining rule-based models such as cause-effect extraction with deep-learning models such as few-shot Question

generation and semantic similarity. Our approach identifies correct and incorrect/missing components in student answers using cause-effect analysis and BERT Transformer. The few-shot Question Generation and re-ranker model then generates questions to help improve student answer. Our model vastly outperforms both simple and strong baselines on student learning gains by a large margin on the Korbit ITS.

One area of future research is to design personalizing feedback for non cause-effect exercises. Another idea is to show multiple feedback to students and have them evaluate it either explicitly or implicitly by trying to answer the question-based feedback. This training signal can be used to further improve the feedback model using active learning.

Bibliography

[1] Christoph Benzmüller, Helmut Horacek, Ivana Kruijff-Korbayova, Manfred Pinkal, Jörg Siekmann, and Magdalena Wolska. Natural language dialog with a tutor system for mathematical proofs. In *Cognitive Systems*, pages 1–14. Springer, 2007.

[2] Paul Blayney and Mark Freeman. Automated formative feedback and summative assessment using individualised spreadsheet assignments. *Australasian Journal of Educational Technology*, 20(2), 2004.

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *NIPS 2020*.

[4] Mengyun Cao, Xiaoping Sun, and Hai Zhuge. The role of cause-effect link within scientific paper. In *2016 12th International Conference on Semantics, Knowledge and Grids (SKG)*, pages 32–39. IEEE, 2016.

[5] Guanliang Chen, Jie Yang, Claudia Hauff, and Geert-Jan Houben. Learningq: a large-scale

dataset for educational question generation. In *AAAI Conference on Web and Social Media*, 2018.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

[7] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation. *NIPS*, 2019.

[8] Xinya Du, Junru Shao, and Claire Cardie. Learning to ask: Neural question generation for reading comprehension. In *ACL*, 2017.

[9] Manaal Faruqui and Dipanjan Das. Identifying well-formed natural language questions. In *EMNLP*, 2018.

[10] Markus Freitag and Yaser Al-Onaizan. Beam search strategies for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 56–60, 2017.

[11] Matt Grenander, Robert Belfer, Ekaterina Kochmar, Iulian V Serban, François St-Hilaire, and Jackie CK Cheung. Deep discourse analysis for generating personalized feedback in intelligent tutor systems. In *The 11th Symposium on Educational Advances in Artificial Intelligence*, 2021.

[12] Xinting Huang, Jianzhong Qi, Yu Sun, and Rui Zhang. Latent reasoning for low-resource question generation. In *Findings of ACL*, 2021.

[13] Junmo Kang, Haritz Puerto San Roman, et al. Let me know what to ask: Interrogative-word-aware question generation. In *2nd Workshop on Machine Reading for Question Answering*, 2019.

[14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[15] Tassilo Klein and Moin Nabi. Learning to answer by learning to ask: Getting the best of gpt-2 and bert worlds. *arXiv preprint arXiv:1911.02365*, 2019.

[16] Ekaterina Kochmar, Dung Do Vu, Robert Belfer, Varun Gupta, Iulian Vlad Serban, and Joelle Pineau. Automated data-driven generation of personalized pedagogical interventions in intelligent tutoring systems. *International Journal of Artificial Intelligence in Education*, 2021.

[17] Devang Kulshreshtha, Robert Belfer, Iulian Vlad Serban, and Siva Reddy. Back-training excels self-training at unsupervised domain adaptation of question generation and passage retrieval. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7064–7078, 2021.

[18] Matthew Lamm, Jennimaria Palomaki, Chris Alberti, Daniel Andor, Eunsol Choi, Livio Baldini

Soares, and Michael Collins. Qed: A framework and dataset for explanations in question answering. *Transactions of the Association for Computational Linguistics*, 9:790–806, 2021.

[19] J Richard Landis and Gary G Koch. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174, 1977.

[20] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*, 2020.

[21] Siyao Li, Deren Lei, Pengda Qin, and William Yang Wang. Deep reinforcement learning with distributional semantic rewards for abstractive summarization. *arXiv preprint arXiv:1909.00141*, 2019.

[22] Ming Liu, Yi Li, Weiwei Xu, and Li Liu. Automated essay feedback generation and its impact on revision. *IEEE Transactions on Learning Technologies*, 10(4):502–513, 2016.

[23] Maxim Makatchev, Pamela W Jordan, Umarani Pappuswamy, and Kurt VanLehn. Representation and reasoning for deeper natural language understanding in a physics tutoring system. *AAAI*, 2011.

[24] Jessica McBroom, Irena Koprinska, and Kalina Yacef. A survey of automated programming hint generation: The hints framework. *ACM Computing Surveys (CSUR)*, 54(8):1–27, 2021.

[25] Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela.

Adversarial nli: A new benchmark for natural language understanding. *arXiv preprint arXiv:1910.14599*, 2019.

[26] Florian Obermüller, Ute Heuer, and Gordon Fraser. Guiding next-step hint generation using automated tests. In *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1*, pages 220–226, 2021.

[27] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002.

[28] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67, 2020.

[29] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In *EMNLP*, 2016.

[30] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *EMNLP*, 2019.

[31] Sascha Rothe, Joshua Maynez, and Shashi Narayan. A thorough evaluation of task-specific pretraining for summarization. In *EMNLP*, 2021.

[32] Lin CY ROUGE. A package for automatic evaluation of summaries. In *Proceedings of Workshop on Text Summarization of ACL, Spain*, 2004.

[33] Megha Srivastava and Noah Goodman. Question generation for adaptive education. In *ACL*, 2021.

[34] Francois St-Hilaire, Nathan Burns, Robert Belfer, Muhammad Shayan, Ariella Smofsky, Dung Do Vu, Antoine Frau, Joseph Potochny, Farid Faraji, Vincent Pavero, et al. A comparative study of learning outcomes for online learning platforms. In *International Conference on Artificial Intelligence in Education*, pages 331–337. Springer, 2021.

[35] Francois St-Hilaire, Dung Do Vu, Antoine Frau, Nathan Burns, Farid Faraji, Joseph Potochny, Stephane Robert, Arnaud Roussel, Selene Zheng, Taylor Glazier, et al. A new era: Intelligent tutoring systems will transform online learning for millions. *arXiv preprint arXiv:2203.03724*, 2022.

[36] John C. Stamper, Michael Eagle, Tiffany Barnes, and Marvin Croy. Experimental Evaluation of Automatic Hint Generation for Logic Tutor. *International Journal of Artificial Intelligence in Education*, 22(1-2):3–17, 2013.

[37] Inigo Jauregi Unanue, Jacob Parnell, and Massimo Piccardi. Berttune: Fine-tuning neural machine translation with bertscore. *arXiv preprint arXiv:2106.02208*, 2021.

[38] Étienne Wenger. *Artificial Intelligence and Tutoring Systems*. Los Altos, CA: Morgan Kaufmann, 1987.

[39] Jianxing Yu, Wei Liu, Shuang Qiu, Qinliang Su, Kai Wang, Xiaojun Quan, and Jian Yin. Low-resource generation of multi-hop reasoning questions. In *ACL*, 2020.

[40] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.

[41] Yao Zhao, Xiaochuan Ni, Yuanyuan Ding, and Qifa Ke. Paragraph-level neural question generation with maxout pointer and gated self-attention networks. In *EMNLP*, 2018.

Authors' addresses

Devang Kulshreshtha School of Computer Science, Mila/McGill University, devang.kulshreshtha@mila.quebec.

Muhammad Shayan Korbit Technologies Montreal, shayan@korbit.ai

Robert Belfer Korbit Technologies Montreal, robert@korbit.ai.

Siva Reddy School of Computer Science, Mila/McGill University, siva.reddy@mila.quebec.

Iulain Vlad Serban Korbit Technologies Montreal, iulian@korbit.ai.

Ekaterina Kochmar University of Bath UK, Korbit Technologies Montreal, ekaterina@korbit.ai.

Appendix

4.A Question Generation Model Training Details

All three models - *T5-QG, BART-QG, BART-ML-QG* are trained for 5 epochs with learning rate of $1e-5$ and batch size of 8. For optimization we use Adam [14] with $\beta_1 = 0.9, \beta_2 = 0.999$. The input and output sequence length is padded to 512 and 150 tokens respectively. For generation we use beam search decoding [10] with number of beams set to 3. The initial checkpoint for the models can be found at - T5-QG[3], BART-QG[4], BART-ML-QG[5]. The hint-assisted QA models - $\theta_{QA}, \theta_{HG}, \theta_{HQA}$ are trained using same configurations and vanilla BART checkpoint[4].

[3] https://huggingface.co/docs/transformers/v4.18.0/en/model_doc/t5#transformers.T5ForConditionalGeneration

[4] https://huggingface.co/docs/transformers/v4.18.0/en/model_doc/bart#transformers.BartForConditionalGeneration

[5] https://huggingface.co/McGill-NLP/bart-qg-mlquestions-backtraining

265 1 1 1 1 {&C (,/;/./–) (&AND) so (-far) (,) &R} [It was not long before he saw me looking at him, and so he began to move.]
35 1 1 1 1 {&C (,/;/./–) (&AND) thus &R} [Japan will be on the dark side of the Earth, and thus will lack the warming influence of the Sun.
9 1 1 1 1 {&C (,/;/./–) consequently (,) &R} [Her mother was seriously ill. Consequently, she left school.]
3 1 1 1 1 {&C (,/;/./–) (&AND) as a (&ADJ) result (,) &R} [All singers kept together. As a result, their performance was successful.]

Table 4.B.1: Examples of the designed cause-effect patterns [4].

4.B Cause-Effect Relation Extraction

We use off-the-shelf cause-effect extractor package [6] as described in Cao et al. [4]. The pattern based discovery algorithm consists of two main steps-

1. *Designing Linguistic patterns for Identifying Causal Relations:* Based on the types of causal expressions, a set of linguistic patterns is constructed that will be used to identify causal relations within a sentence. Each pattern is basically a template for expressing cause and effect, and is equivalent to a finite state transition network.

2. *Matching sentences with patterns to get candidates:* A sentence is matched with each pattern, a match means potential cause-effect relation is present.

Table 4.B.1 enlists some of the designed cause-effect patterns. Refer to Cao et al. [4] for more details on all patterns and above steps.

[6] `https://github.com/Angela7126/CE_extractor--Patterns_Based`

4.B.1 Finding most similar reference solution

The hint generation procedure (Algorithm 2) requires finding most similar reference solution s_r to student solution s_s as a first step. This is achieved by comparing similarity of each $s_r \in \mathscr{R}$ with s_s using BERTScore. As an alternate approach, we also tried to first decompose each $s_r \in \mathscr{R}$ into their cause-effect constituents $\{c_r, e_r\}$ and compare the constituents with student solution constituents $\{c_s, e_s\}$ to find most similar reference solution. However, manual run on a bunch of sentences revealed former method to be better at identifying most similar reference solution.

Chapter 5

Conclusion

5.1 Contribution to Original Knowledge

In this thesis, we have detailed various works completed towards leveraging Question Generation for improving Intelligent Tutoring Systems (ITS), including the development of new dataset for evaluating Educational QG models (Paper-I), a new algorithm for domain adaptation of QG which vastly outperforms existing methods (Paper-I), and a question-based personalized feedback generation model for improving student learning gains an ITS (Paper-II).

Specifically, above contributions correspond to fulfilling these research directions:

Resources for Educational QG: How can we develop new datasets for training and evaluation of Question Generation models for education domain? Such datasets can also be used to foster research in Unsupervised Domain Adaptation (UDA).

Most datasets such as SQuAD [88], NewsQA [112], TriviaQA [48], NarrativeQA [55] etc. were developed for reading comprehension materials. These datasets are however not suitable for educational content as they belong to generic domain and consist of mainly factoid questions.

We developed a new domain-adaptation dataset for Question Generation called MLQuestions [58] containing 35K unaligned questions, 50K unaligned passages, and 3K aligned question-passage pairs. Our model generated high-quality questions for target (educational) domain compared to models without domain adaptation.

Algorithms/Models for Educational QG: How can we develop algorithms for Question Generation that can generalize/perform well to educational domain?

Early work builds educational QG models by training Seq2Seq models on educational QG datasets [21]. There is no direct work in our knowledge for domain adaptation in QG.

We develop the back-training algorithm for UDA of Question Generation and Passage Retrieval which vastly outperformed the popular self-training [124] algorithm by a mean improvement of 7.8 BLEU4 points on generation, and 17.6% top-20 retrieval accuracy education and medical domain. Our algorithm significantly reduced the gap between the target domain and synthetic data distribution, and reduced model overfitting to the source domain.

Personalized Feedback Generation Models: How can we build personalized question-based feedback systems in ITS? The goal here is to pinpoint correct and incorrect/missing components in student answers, and provide feedback that guide students towards improving their answer.

Existing work on generating hints in an Intelligent Tutoring System (ITS) explores mostly manual, static, and non-personalized feedback.

We explore automatically generated questions as personalized feedback in an ITS. Our model generated feedback which pinpoints correct and incorrect/missing components in student answers, and provide feedback in the form of natural language questions that guide students towards improving their answer. Our model vastly outperforms both simple and strong baselines on student learning gains by 30% when tested on a real dialogue-based ITS.

5.2 Limitations and Future Directions

Despite the contributions described from the works presented, a number of crucial problems still remain unsolved in advancing educational ITS using Question Generation.

Until now, the questions generated by educational QG models require single step of reasoning. Potential future directions may include multi-hop *educational* QG by leveraging recent research in multi-hop QG [39, 108].

Additionally, the MLQuestions is only an evaluation dataset however creating a training dataset will likely improve quality of educational QG models, since supervised algorithms currently vastly outperform unsupervised algorithms in NLP.

Secondly, the limited training data in languages other than English hinders development of QG models. Steps towards addressing this could include multilingual QG by leveraging research in multilingual NMT [1, 57] as well as collecting training data specific to language of interest.

The personalized feedback generation system described in our paper works only for a specific category of exercises. Steps towards personalizing automated feedback models for exercises involving mathematical equations by leveraging recent research in symbolic language models [25] would be a great contribution.

Additionally, the current level of personalization is based only on the context of the ongoing conversation. Incorporating student past history (e.g. student knowledge level, experience on platform etc.) can be used additionally to further personalize the feedback model. Similar features have been used for personalizing questions for individual students in educational ITS [104].

Finally, an interesting research direction is to show multiple feedback to students and have them evaluate it either explicitly or implicitly by trying to answer the question-based feedback. This training signal can be used to further improve the feedback model using active learning.

Bibliography

[1] Roee Aharoni, Melvin Johnson, and Orhan Firat. Massively multilingual neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3874–3884, 2019.

[2] Patricia Albacete, Pamela Jordan, Sandra Katz, Irene-Angelica Chounta, and Bruce M McLaren. The impact of student model updates on contingent scaffolding in a natural-language tutoring system. In *International conference on artificial intelligence in education*, pages 37–47. Springer, 2019.

[3] Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. Synthetic QA corpora generation with roundtrip consistency. In *ACL*, 2019.

[4] Joanne Anania. The Influence of Instructional Conditions on Student Learning and Achievement. *Evaluation in Education: An International Review Series*, 7(1):3–76, 1983.

[5] John R Anderson, C Franklin Boyle, and Brian J Reiser. Intelligent tutoring systems. *Science*, 228(4698):456–462, 1985.

[6] Mikel Artetxe, Gorka Labaka, and Eneko Agirre. Unsupervised statistical machine translation. In *EMNLP*, 2018.

[7] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[8] Harry P Bahrick, Lorraine E Bahrick, Audrey S Bahrick, and Phyllis E Bahrick. Maintenance of foreign language vocabulary and the spacing effect. *Psychological Science*, 4(5):316–321, 1993.

[9] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, 2005.

[10] Yoshua Bengio. Deep learning of representations for unsupervised and transfer learning. In *ICML workshop on unsupervised and transfer learning*, 2012.

[11] Christoph Benzmüller, Helmut Horacek, Ivana Kruijff-Korbayova, Manfred Pinkal, Jörg Siekmann, and Magdalena Wolska. Natural language dialog with a tutor system for mathematical proofs. In *Cognitive Systems*, pages 1–14. Springer, 2007.

[12] Paul Blayney and Mark Freeman. Automated formative feedback and summative assessment

using individualised spreadsheet assignments. *Australasian Journal of Educational Technology*, 20(2), 2004.

[13] Benjamin S Bloom. The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational researcher*, 13(6):4–16, 1984.

[14] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *CoNLL*, 1998.

[15] Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. In *EMNLP*, 2015.

[16] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *NIPS 2020*.

[17] Arthur Joseph Burke. *Students' potential for learning contrasted under tutorial and group approaches to instruction*. PhD thesis, University of Chicago, Joseph Regenstein Library, Department of Photoduplication, 1983.

[18] Mengyun Cao, Xiaoping Sun, and Hai Zhuge. The role of cause-effect link within scientific paper. In *2016 12th International Conference on Semantics, Knowledge and Grids (SKG)*, pages 32–39. IEEE, 2016.

[19] Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. Semeval-2017

task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *SemEval*, 2017.

[20] Ying-Hong Chan and Yao-Chung Fan. Bert for question generation. In *INLG*, 2019.

[21] Guanliang Chen, Jie Yang, Claudia Hauff, and Geert-Jan Houben. Learningq: a large-scale dataset for educational question generation. In *AAAI Conference on Web and Social Media*, 2018.

[22] Min Chi, Kenneth R Koedinger, Geoffrey J Gordon, Pamela W Jordan, and Kurt VanLehn. Instructional factors analysis: A cognitive model for multiple instructional interventions. *EDM*, 2011:61–70, 2011.

[23] Yu-An Chung, Hung-Yi Lee, and James Glass. Supervised and unsupervised transfer learning for question answering. In *NAACL*, 2018.

[24] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys (Csur)*, 2008.

[25] David Demeter and Doug Downey. Just add functions: A neural-symbolic language model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7634–7642, 2020.

[26] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

[27] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation. *NIPS*, 2019.

[28] Xinya Du, Junru Shao, and Claire Cardie. Learning to ask: Neural question generation for reading comprehension. In *ACL*, 2017.

[29] Nan Duan, Duyu Tang, Peng Chen, and Ming Zhou. Question generation for question answering. In *EMNLP*, 2017.

[30] Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. Understanding back-translation at scale. In *EMNLP*, 2018.

[31] Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation. In *ACL*, 2018.

[32] Manaal Faruqui and Dipanjan Das. Identifying well-formed natural language questions. In *EMNLP*, 2018.

[33] Juan M Fernández-Luna, Juan F Huete, Andrew MacFarlane, and Efthimis N Efthimiadis. Teaching and learning in information retrieval. *Information Retrieval*, 2009.

[34] Jeremiah T Folsom-Kovarik, Sae Schatz, and Denise Nicholson. Plan ahead: Pricing its learner models. In *Proceedings of the 19th Behavior Representation in Modeling & Simulation (BRIMS) Conference*, pages 47–54, 2010.

[35] Markus Freitag and Yaser Al-Onaizan. Beam search strategies for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 56–60, 2017.

[36] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.

[37] Arthur C Graesser, Kurt VanLehn, Carolyn P Rosé, Pamela W Jordan, and Derek Harter. Intelligent tutoring systems with conversational dialogue. *AI magazine*, 22(4):39–39, 2001.

[38] Matt Grenander, Robert Belfer, Ekaterina Kochmar, Iulian V Serban, François St-Hilaire, and Jackie CK Cheung. Deep discourse analysis for generating personalized feedback in intelligent tutor systems. In *The 11th Symposium on Educational Advances in Artificial Intelligence*, 2021.

[39] Deepak Gupta, Hardik Chauhan, Ravi Tej Akella, Asif Ekbal, and Pushpak Bhattacharyya. Reinforced multi-task approach for multi-hop question generation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2760–2775, 2020.

[40] Çağlar Gulçehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. Pointing the unknown words, 2016.

[41] Vrindavan Harrison and Marilyn Walker. Neural generation of diverse questions using answer focus, contextual and linguistic features. *INLG 2018*, page 296, 2018.

[42] Yulan He and Deyu Zhou. Self-training from labeled features for sentiment analysis. *Information Processing & Management*, 47(4), 2011.

[43] Michael Heilman and Noah A Smith. Good question! statistical ranking for question generation. In *NAACL*, 2010.

[44] Stefan Hrastinski, Stefan Stenbom, Simon Benjaminsson, and Malin Jansson. Identifying and exploring the effects of different types of tutor questions in individual online synchronous tutoring in mathematics. *Interactive Learning Environments*, 0(0):1–13, 2019. doi: 10.1080/10494820.2019.1583674. URL `https://doi.org/10.1080/10494820.2019.1583674`.

[45] Xinting Huang, Jianzhong Qi, Yu Sun, and Rui Zhang. Latent reasoning for low-resource question generation. In *Findings of ACL*, 2021.

[46] Gregory Hume, Joel Michael, Allen Rovick, and Martha Evens. Hinting as a tactic in one-on-one tutoring. *The Journal of the Learning Sciences*, 5(1):23–47, 1996.

[47] Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. Pubmedqa: A dataset for biomedical research question answering. In *EMNLP*, 2019.

[48] Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th*

Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1601–1611, 2017.

[49] Junmo Kang, Haritz Puerto San Roman, et al. Let me know what to ask: Interrogative-word-aware question generation. In *2nd Workshop on Machine Reading for Question Answering*, 2019.

[50] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *EMNLP*, 2020.

[51] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[52] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[53] Tassilo Klein and Moin Nabi. Learning to answer by learning to ask: Getting the best of gpt-2 and bert worlds. *arXiv preprint arXiv:1911.02365*, 2019.

[54] Ekaterina Kochmar, Dung Do Vu, Robert Belfer, Varun Gupta, Iulian Vlad Serban, and Joelle Pineau. Automated data-driven generation of personalized pedagogical interventions in intelligent tutoring systems. *International Journal of Artificial Intelligence in Education*, 2021.

[55] Tomáš Kočiskỳ, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328, 2018.

[56] Kenneth R Koedinger, Albert Corbett, et al. *Cognitive tutors: Technology bringing learning sciences to the classroom*. na, 2006.

[57] Sneha Kudugunta, Ankur Bapna, Isaac Caswell, and Orhan Firat. Investigating multilingual nmt representations at scale. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1565–1575, 2019.

[58] Devang Kulshreshtha, Robert Belfer, Iulian Vlad Serban, and Siva Reddy. Back-training excels self-training at unsupervised domain adaptation of question generation and passage retrieval. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7064–7078, 2021.

[59] Vishwajeet Kumar, Kireeti Boorla, Yogesh Meena, Ganesh Ramakrishnan, and Yuan-Fang Li. Automating reading comprehension by generating question and answer pairs. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 335–348. Springer, 2018.

[60] Vishwajeet Kumar, Ganesh Ramakrishnan, and Yuan-Fang Li. A framework for automatic question generation from text using deep reinforcement learning. *arXiv preprint arXiv:1808.04961*, 2018.

[61] Vishwajeet Kumar, Ganesh Ramakrishnan, and Yuan-Fang Li. Putting the horse before the cart: A generator-evaluator framework for question generation from text. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 812–821, 2019.

[62] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *TACL*, 2019.

[63] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. Race: Large-scale reading comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, 2017.

[64] Matthew Lamm, Jennimaria Palomaki, Chris Alberti, Daniel Andor, Eunsol Choi, Livio Baldini Soares, and Michael Collins. Qed: A framework and dataset for explanations in question answering. *Transactions of the Association for Computational Linguistics*, 9: 790–806, 2021.

[65] J Richard Landis and Gary G Koch. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174, 1977.

[66] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*, 2020.

[67] Siyao Li, Deren Lei, Pengda Qin, and William Yang Wang. Deep reinforcement learning with distributional semantic rewards for abstractive summarization. *arXiv preprint arXiv:1909.00141*, 2019.

[68] Zhenghua Li, Min Zhang, and Wenliang Chen. Ambiguity-aware ensemble training for semi-supervised dependency parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 457–467, Baltimore, Maryland, June 2014. Association for Computational Linguistics. doi: 10.3115/v1/P14-1043. URL https://www.aclweb.org/anthology/P14-1043.

[69] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, 2004.

[70] David Lindberg, Fred Popowich, John Nesbit, and Phil Winne. Generating natural language questions to support learning on-line. In *14th European Workshop on Natural Language Generation*, 2013.

[71] Ming Liu, Yi Li, Weiwei Xu, and Li Liu. Automated essay feedback generation and its impact on revision. *IEEE Transactions on Learning Technologies*, 10(4):502–513, 2016.

[72] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[73] Maxim Makatchev, Pamela W Jordan, Umarani Pappuswamy, and Kurt VanLehn. Representation and reasoning for deeper natural language understanding in a physics tutoring system. *AAAI*, 2011.

[74] Jessica McBroom, Irena Koprinska, and Kalina Yacef. A survey of automated programming hint generation: The hints framework. *ACM Computing Surveys (CSUR)*, 54(8):1–27, 2021.

[75] Mary L McHugh. Interrater reliability: the kappa statistic. *Biochemia medica*, 2012.

[76] Robert J Menner. The conflict of homonyms in english. *Language*, 1936.

[77] Shlok Kumar Mishra, Pranav Goel, Abhishek Sharma, Abhyuday Jagannatha, David Jacobs, and Hal Daume. Towards automatic generation of questions from long answers. *arXiv preprint arXiv:2004.05109*, 2020.

[78] Nasrin Mostafazadeh, Ishan Misra, Jacob Devlin, Margaret Mitchell, Xiaodong He, and Lucy Vanderwende. Generating natural questions about an image. In *ACL*, 2016.

[79] Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial nli: A new benchmark for natural language understanding. *arXiv preprint arXiv:1910.14599*, 2019.

[80] R Nielsen. Question generation: Proposed challenge tasks and their evaluation. In *Workshop on the Question Generation Shared Task and Evaluation Challenge, Arlington, Virginia*, 2008.

[81] Rodney D Nielsen, Jason Buckingham, Gary Knoll, Ben Marsh, and Leysia Palen. A taxonomy of questions for question generation. In *Workshop on the Question Generation Shared Task and Evaluation Challenge*, 2008.

[82] Benjamin D Nye, Arthur C Graesser, and Xiangen Hu. AutoTutor and family: A review of 17 years of natural language tutoring. *International Journal of Artificial Intelligence in Education*, 24(4):427–469, 2014.

[83] Florian Obermüller, Ute Heuer, and Gordon Fraser. Guiding next-step hint generation using automated tests. In *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1*, pages 220–226, 2021.

[84] Andrew M Olney and Whitney L Cade. Authoring intelligent tutoring systems using human computation: designing for intrinsic motivation. In *International conference on augmented cognition*, pages 628–639. Springer, 2015.

[85] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002.

[86] Michael Prince. Does active learning work? a review of the research. *Journal of engineering education*, 93(3):223–231, 2004.

[87] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena,

Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67, 2020.

[88] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In *EMNLP*, 2016.

[89] Alan Ramponi and Barbara Plank. Neural unsupervised domain adaptation in NLP—A survey. In *COLING*, 2020.

[90] Revanth Gangi Reddy, Bhavani Iyer, Md Arafat Sultan, Rong Zhang, Avi Sil, Vittorio Castelli, Radu Florian, and Salim Roukos. End-to-end qa on covid-19: Domain adaptation with synthetic training. *arXiv preprint arXiv:2012.01414*, 2020.

[91] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *EMNLP*, 2019.

[92] Stephen Robertson and Hugo Zaragoza. *The probabilistic relevance framework: BM25 and beyond*. Now Publishers Inc, 2009.

[93] Sascha Rothe, Joshua Maynez, and Shashi Narayan. A thorough evaluation of task-specific pretraining for summarization. In *EMNLP*, 2021.

[94] Lin CY ROUGE. A package for automatic evaluation of summaries. In *Proceedings of Workshop on Text Summarization of ACL, Spain*, 2004.

[95] Sebastian Ruder, Matthew E Peters, Swabha Swayamdipta, and Thomas Wolf. Transfer learning in natural language processing. In *NAACL: Tutorials*, 2019.

[96] Vasile Rus and James Lester. The 2nd workshop on question generation. In *Artificial Intelligence in Education*, pages 808–808. IOS Press, 2009.

[97] Vasile Rus, Zhiqiang Cai, and Art Graesser. Question generation: Example of a multi-year evaluation campaign. *Proc WS on the QGSTEC*, 2008.

[98] Vasile Rus, Brendan Wyse, Paul Piwek, Mihai Lintean, Svetlana Stoyanchev, and Cristian Moldovan. Overview of the first question generation shared task evaluation challenge. In *Proceedings of the Third Workshop on Question Generation*, pages 45–57, 2010.

[99] Vasile Rus, Brendan Wyse, Paul Piwek, Mihai Lintean, Svetlana Stoyanchev, and Christian Moldovan. Question generation shared task and evaluation challenge–status report. In *Proceedings of the 13th European Workshop on Natural Language Generation*, pages 318–320, 2011.

[100] Vasile Rus, Brendan Wyse, Paul Piwek, Mihai Lintean, Svetlana Stoyanchev, and Cristian Moldovan. A detailed account of the first question generation shared task evaluation challenge. *Dialogue & Discourse*, 3(2):177–204, 2012.

[101] Mrinmaya Sachan and Eric Xing. Self-training for jointly learning to ask and answer questions. In *NAACL*, 2018.

[102] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[103] Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. In *ACL*, 2016.

[104] Megha Srivastava and Noah Goodman. Question generation for adaptive education. In *ACL*, 2021.

[105] Francois St-Hilaire, Nathan Burns, Robert Belfer, Muhammad Shayan, Ariella Smofsky, Dung Do Vu, Antoine Frau, Joseph Potochny, Farid Faraji, Vincent Pavero, et al. A comparative study of learning outcomes for online learning platforms. In *International Conference on Artificial Intelligence in Education*, pages 331–337. Springer, 2021.

[106] Francois St-Hilaire, Dung Do Vu, Antoine Frau, Nathan Burns, Farid Faraji, Joseph Potochny, Stephane Robert, Arnaud Roussel, Selene Zheng, Taylor Glazier, et al. A new era: Intelligent tutoring systems will transform online learning for millions. *arXiv preprint arXiv:2203.03724*, 2022.

[107] John C. Stamper, Michael Eagle, Tiffany Barnes, and Marvin Croy. Experimental Evaluation of Automatic Hint Generation for Logic Tutor. *International Journal of Artificial Intelligence in Education*, 22(1-2):3–17, 2013.

[108] Dan Su, Yan Xu, Wenliang Dai, Ziwei Ji, Tiezheng Yu, and Pascale Fung. Multi-hop

question generation with graph convolutional network. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4636–4647, 2020.

[109] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014.

[110] Richard S Sutton, Andrew G Barto, et al. Introduction to reinforcement learning. 1998.

[111] Duyu Tang, Nan Duan, Tao Qin, Zhao Yan, and Ming Zhou. Question answering and question generation as dual tasks. *arXiv preprint arXiv:1706.02027*, 2017.

[112] Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. Newsqa: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200, 2017.

[113] Nicola Ueffing. Self-training for machine translation. In *NIPS workshop on Machine Learning for Multilingual Information Access*, 2006.

[114] Inigo Jauregi Unanue, Jacob Parnell, and Massimo Piccardi. Berttune: Fine-tuning neural machine translation with bertscore. *arXiv preprint arXiv:2106.02208*, 2021.

[115] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[116] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *ICML*, 2008.

[117] Xiaojun Wan. Co-training for cross-lingual sentiment classification. In *ACL*, 2009.

[118] Tong Wang, Xingdi Yuan, and Adam Trischler. A joint model for question answering and question generation. *arXiv preprint arXiv:1706.01450*, 2017.

[119] Zichao Wang, Andrew S Lan, Weili Nie, Andrew E Waters, Phillip J Grimaldi, and Richard G Baraniuk. Qg-net: a data-driven question generation model for educational content. In *Proceedings of the Fifth Annual ACM Conference on Learning at Scale*, pages 1–10, 2018.

[120] David Weiss, Chris Alberti, Michael Collins, and Slav Petrov. Structured training for neural network transition-based parsing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 323–333, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1032. URL https://www.aclweb.org/anthology/P15-1032.

[121] Étienne Wenger. *Artificial Intelligence and Tutoring Systems*. Los Altos, CA: Morgan Kaufmann, 1987.

[122] Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *NAACL*, 2018.

[123] Yuxi Xie, Liangming Pan, Dongzhe Wang, Min-Yen Kan, and Yansong Feng. Exploring question-specific rewards for generating deep questions. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2534–2546, 2020.

[124] David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *ACL*, 1995.

[125] Jianxing Yu, Wei Liu, Shuang Qiu, Qinliang Su, Kai Wang, Xiaojun Quan, and Jian Yin. Low-resource generation of multi-hop reasoning questions. In *ACL*, 2020.

[126] Xiang Yu, Ngoc Thang Vu, and Jonas Kuhn. Ensemble self-training for low-resource languages: grapheme-to-phoneme conversion and morphological inflection. In *SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, 2020.

[127] Xingdi Yuan, Tong Wang, Çağlar Gulçehre, Alessandro Sordoni, Philip Bachman, Saizheng Zhang, Sandeep Subramanian, and Adam Trischler. Machine comprehension by text-to-text neural question generation. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 15–25, 2017.

[128] Hamed Zamani, Susan Dumais, Nick Craswell, Paul Bennett, and Gord Lueck. Generating clarifying questions for information retrieval. In *Proceedings of The Web Conference 2020*, pages 418–428, 2020.

[129] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.

[130] Zhirui Zhang, Shuo Ren, Shujie Liu, Jianyong Wang, Peng Chen, Mu Li, Ming Zhou, and Enhong Chen. Style transfer as unsupervised machine translation. *arXiv preprint arXiv:1808.07894*, 2018.

[131] Han Zhao, Remi Tachet Des Combes, Kun Zhang, and Geoffrey Gordon. On learning invariant representations for domain adaptation. In *ICML*, 2019.

[132] Yao Zhao, Xiaochuan Ni, Yuanyuan Ding, and Qifa Ke. Paragraph-level neural question generation with maxout pointer and gated self-attention networks. In *EMNLP*, 2018.

[133] Xiaojin Zhu. Semi-supervised learning literature survey. Technical report, Computer Sciences, University of Wisconsin-Madison, 2005.