Fluent Response Generation for Conversational Question Answering

Ashutosh Baheti, Alan Ritter
Computer Science and Engineering
Ohio State University
{baheti.3, ritter.1492}osu.edu

Kevin Small
Amazon Alexa
smakevin@amazon.com

Abstract

Question answering (QA) is an important aspect of open-domain conversational agents, garnering specific research focus in the conversational QA (ConvQA) subtask. One notable limitation of recent ConvQA efforts is the response being answer span extraction from the target corpus, thus ignoring the natural language generation (NLG) aspect of high-quality conversational agents. In this work, we propose a method for situating QA responses within a SEQ2SEQ NLG approach to generate fluent grammatical answer responses while maintaining correctness. From a technical perspective, we use data augmentation to generate training data for an end-to-end system. Specifically, we develop Syntactic Transformations (STs) to produce question-specific candidate answer responses and rank them using a BERT-based classifier (Devlin et al., 2019). Human evaluation on SQuAD 2.0 data (Rajpurkar et al., 2018) demonstrate that the proposed model outperforms baseline CoQA and QuAC models in generating *conversational* responses. We further show our model's scalability by conducting tests on the CoQA dataset.[1]

1 Introduction

Factoid question answering (QA) has recently enjoyed rapid progress due to the increased availability of large crowdsourced datasets (e.g., SQuAD (Rajpurkar et al., 2016), MS MARCO (Bajaj et al., 2016), Natural Questions (Kwiatkowski et al., 2019)) for training neural models and the significant advances in pre-training contextualized representations using massive text corpora (e.g., ELMo (Peters et al., 2018), BERT (Devlin et al., 2019)). Building on these successes, recent work examines *conversational* QA (ConvQA) systems capable of interacting with users over multiple turns. Large crowdsourced ConvQA datasets (e.g., CoQA (Reddy et al., 2019), QuAC (Choi et al., 2018)) consist of dialogues between crowd workers who are prompted to ask and answer a sequence of questions regarding a source document. Although these ConvQA datasets support multi-turn QA interactions, the responses have mostly been limited to extracting text spans from the source document and do not readily support abstractive answers (Yatskar, 2019). While responses copied directly from a Wikipedia article can provide a correct answer to a user question, they do not sound natural in a conversational setting. To address this challenge, we develop SEQ2SEQ models that generate fluent and informative answer responses to conversational questions.

To obtain data needed to train these models, rather than constructing yet-another crowdsourced QA dataset, we transform the answers from an existing QA dataset into fluent responses via data augmentation. Specifically, we synthetically generate supervised training data by converting questions and associated extractive answers from a SQuAD-like QA dataset into fluent responses via *Syntactic Transformations* (STs). These STs over-generate a large set of candidate responses from which a BERT-based classifier selects the best response as shown in the top half of Figure 1.

While over-generation and selection generates fluent responses in many cases, the brittleness of the off-the-shelf parsers and the syntatic transformation rules prevent direct use in cases that are not well-covered. To mitigate this limitation, we generate a new augmented training dataset using the best response classifier that is used to train end-to-end response generation models based on Pointer-Generator Networks (PGN) (See et al., 2017) and pre-trained Transformers using large amounts of dialogue data, DialoGPT (D-GPT) (Zhang et al., 2019). In §3.2 and §3.3, we empirically demon-

[1]The code and data are available at https://github.com/abaheti95/QADialogSystem.

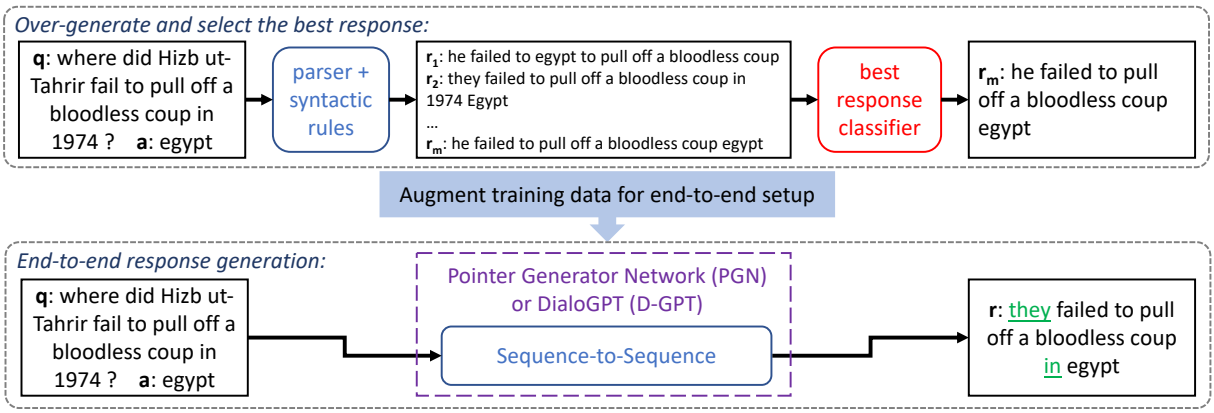

Figure 1: Overview of our method of generating conversational responses for a given QA. In the first method, the *Syntactic Transformations* (STs) over-generate a list of responses (good and bad) using the question's parse tree and the *best response classifier* selects the most suitable response from the list. Our second method uses this pipeline to augment training data for training a SEQ2SEQ networks PGN or D-GPT (§3.1). The final SEQ2SEQ model is end-to-end, scalable, easier to train, and performs better than the first method exclusively.

strate that our proposed NLG models are capable of generating fluent, abstractive answers on both SQuAD 2.0 and CoQA.

2 Generating Fluent QA Responses

In this section, we describe our approach for constructing a corpus of questions and answers that supports fluent answer generation (top half of Figure 1). We use the framework of **overgenerate and rank** previously used in the context of question generation (Heilman and Smith, 2010). We first **overgenerate** answer responses for QA pairs using STs in §2.1. We then **rank** these responses from best to worst using the response classification models described in §2.2. Later in §3, we describe how we augment existing QA datasets with fluent answer responses using STs and a best response classifier. This augmented QA dataset is used for training the PGN and Transformer models.

2.1 Syntactic Transformations (STs)

The first step is to apply the Syntactic Transformations (STs) to the question's parse tree along with the expert answer phrase to produce multiple candidate responses. For the STs to work effectively accurate question parses are essential. We use the Stanford English lexparser[2](Klein and Manning, 2003), which is trained on WSJ sections 1-21, QuestionBank (Judge et al., 2006), amongst other corpora. However, this parser still fails to recognize $\sim 20\%$ of the questions (neither SBARQ nor SQ tag is assigned). For such erroneous parse trees, we simply output the expert answer phrase as a single response. The remaining questions are processed via the following transformations to over-generate a list of candidate answers: (1) **Verb modification**: change the tense of the main verb based on the auxiliary verb using SimpleNLG (Gatt and Reiter, 2009); (2) **Pronoun replacement**: substitute the noun phrase with pronouns from a fixed list; (3) **Fixing Preposition and Determiner**: find the preposition and determiner in the question's parse tree that connects to the answer phrase and add all possible prepositions and determiners if missing. (4) **Response Generation**: Using Tregex and Tsurgeon (Levy and Andrew, 2006), compile responses by combining components of all previous steps and the answer phrase. In cases where there are multiple options in steps (2) and (3), the number of options can explode and we use the best response classifier (described below) to winnow. An example ST process is shown in Figure 2.

2.2 Response Classification and Baselines

A classification model selects the best response from the list of ST-generated candidates. Given the training dataset, D, described in §2.3 of n question-answer tuples (q_i, a_i), and their list of corresponding responses, $\{r_{i1}, r_{i2}, ..., r_{im_i}\}$, the goal is to classify each response r_{ij} as bad or good. The probability of the response being good is later used for ranking. We experiment with two different model objectives described below,

Logistic: We assume that the responses for each q_i are independent of each other. The model ($F()$) classifies each response separately and assigns 1 (or 0) if r_{ij} is a good (or bad) response for q_i. The **Logistic** loss is given by $\sum_{i=1}^{n}\sum_{j=1}^{m_i}\log(1+$

[2]https://nlp.stanford.edu/software/parser-faq.html#z

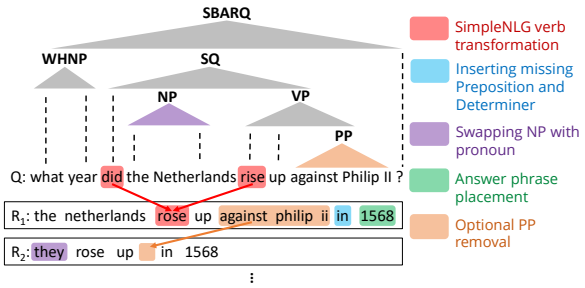

Figure 2: An example of Syntactic Transformations in action. Question: "what year did the Netherlands rise up against Philip II?" Answer: "1568". Using the question's parse tree we: (1) modify the verb "rise" based on the auxiliary verb "did" (red); (2) add missing prepositions and determiners (sky blue); (3) combine the subject and other components with the answer phrase (green) to generate the candidate R_1. In another candidate R_2, we swap the subject with pronoun "they" (purple). Our transformations can also optionally remove Prepositional-Phrases (PP) as shown in R_2 (orange). In the figure, we only show two candidates but in reality the transformations generate many more different candidates, including many implausible ones.

$e^{-y_{ij} * F(q_i, a_i, r_{ij})}$), where y_{ij} is the label for r_{ij}.

Softmax: We will discuss in §2.3 that annotators are expected to miss a few good responses since good and bad answers are often very similar (may only differ by a single preposition or pronoun). Therefore, we explore a ranking objective that calculates errors based on the margin with which incorrect responses are ranked above correct ones (Collins and Koo, 2005). Without loss of generality, we assume r_{i1} to be better than all other responses for (q_i, a_i). Since the model $F()$ should rank r_{i1} higher than all other responses, we use the margin error $M_{ij}(F) = F(q_i, a_i, r_{i1}) - F(q_i, a_i, r_{ij})$ to define the **Softmax** loss as $\sum_{i=1}^{n} \log\left(1 + \sum_{j=2}^{m_i} e^{-M_{ij}(F)}\right)$.

We experiment with the following feature based and neural models with the two loss functions:

Language Model Baseline: The responses are ranked using the normalized probabilities from a 3-gram LM trained on the Gigaword corpus with modified Kneser-Ney smoothing.[3] The response with the highest score is classified as 1 and others as 0.

Linear Model: A linear classifier using features inspired by Heilman and Smith (2010) and Wan et al. (2006), who have implemented similar linear models for other sentence pair classification tasks. Specifically, we use the following features:

[3] http://www.keithv.com/software/giga/

- Length (**Features 1-3**): word length of question q_i, answer-phrase a_i, and response r_{ij}
- WH-word (**Features 4-12**): [0-1 feat.] *what*, *who*, *whom*, *whose*, *when*, *where*, *which*, *why* or *how* is present in the q_i
- Negation (**Features 13**): [0-1 feat.] *no*, *not* or *none* is present in the q_i
- N-gram LM (**Features 14-21**): 2, 3-gram normalized probability and perplexity of q_i and r_{ij}
- Grammar (**Features 22-93**): node counts of q_i and r_{ij} syntactic parse trees
- Word overlap (**Features 94-96**): three features based on fraction of word overlap between q_i and r_{ij}. $precision = \frac{overlap(q_i, r_{ij})}{|q_i|}$, $recall = \frac{overlap(q_i, r_{ij})}{|r_{ij}|}$ and their harmonic mean

Decomposable Attention: We use the sentence pair classifier from (Parikh et al., 2016), referred as the **DA** model. It finds attention based word-alignment of the input pair (premise and hypothesis, in our case question q_i and response r_{ij}) and aggregates it using feedforward networks. Apart from standard vector embeddings, we also experiment with contextualized ELMo (Peters et al., 2018) embedding with the **DA** model using the version implemented in AllenNLP (Gardner et al., 2017).

BERT: Lastly, we use the BERT-Base, Uncased model (Devlin et al., 2019) for sentence pair classification. The model takes question q_i and response r_{ij} separated by the special token `[SEP]` and predicts if the response is suitable or unsuitable.

In some cases, the number of responses generated by the STs for a question could be as high as 5000+. Therefore, when training the **DA** model with pre-trained contextualized embeddings such as ELMo or the **BERT** model in the **Softmax** loss setting, backpropagation requires computing and storing hidden states for 5000+ different responses. To mitigate this issue, we use *strided negative-sampling*. While training, we first separate all the suitable responses from all the remaining unsuitable responses. We then divide all the responses for q_i into smaller batches of K or fewer responses. Each batch comprises one suitable response (randomly chosen) and $K-1$ sampled from the unsuitable responses. To ensure that all unsuitable responses are used at least once during the training, we shuffle them and then create smaller batches by taking strides of $K-1$ size. We use $K = 150$ for **DA**+ELMo and $K = 50$ for **BERT** when training with the **Softmax** loss. At test time, we com-

pute logits on the CPU and normalize across all responses.

2.3 Training Data for Response Classification

In this section, we describe the details of the training, validation and testing data used to develop the *best response classifier* models. To create the supervised data, we choose a sample from the *train-set* of the SQuAD 2.0 dataset (Rajpurkar et al., 2018). SQuAD 2.0 contains human-generated questions and answer spans selected from Wikipedia paragraphs. Before sampling, we remove all the QA pairs which had answer spans > 5 words as they tend to be non-factoid questions and complete sentences in themselves (typically "why" and "how" questions). We also filter out questions that cannot be handled by the parser ($\sim 20\%$ of them had obvious parser errors). After these filtering, we take a sample of 3000 questions and generate their list of responses using STs (1,561,012 total responses).

Next, we developed an annotation task on Amazon Mechanical Turk to select the best responses for the questions. For each question, we ask the annotators to select a response from the list of responses that correctly answers the question, sounds natural, and seems human-like. Since the list of responses for some questions is as long as 5000+, the annotators can't review all of them before selecting the best one. Hence, we implement a search feature within the responses list such that annotators can type in a partial response in the search box to narrow down the options before selection. To make their job easier, we also sorted responses by length. This encouraged annotators to select relatively short responses which we found to be beneficial, as one would prefer an automatic QA system to be terse. To verify that the annotators didn't cheat this annotation design by selecting the first/shortest option, we also test a **Shortest Response Baseline** as another baseline response classifier model, where first/shortest response in the list is selected as suitable.

Each question is assigned 5 annotators. Therefore, there can be at most 5 unique annotated responses for each question. This decreases the recall of the gold truth data (since there can be more than 5 good ways of correctly responding to a question). On the other hand, bad annotators may choose a unique yet suboptimal/incorrect response, which decreases the precision of the gold truth.

After annotating the 3000 questions from SQuAD 2.0 sample, we randomly split the data into 2000 train, 300 validation, and 700 test questions. We refer to this as the SQuAD Gold annotated (**SG**) data. To increase **SG** training data precision, we assign label 1 only to responses that are marked as best by at least two different annotators. Due to this hard constraint, 244 questions from the training data are removed (i.e. the 5 annotators marked 5 unique responses). On the other hand, to increase the recall of the **SG** test and validation sets, we retain all annotations.[4] We assign label 0 to all remaining responses (even if some of them are plausible). The resulting **SG** data split is summarized in Table 1.

	$\#q/\#a$	✓$\#r$	✗$\#r$
Train	1756	2028	796174
Val	300	791	172135
Test	700	1833	182963

Table 1: Statistics of the **SG** training, validation, and test sets curated from the SQuAD 2.0 *training* data. q and a denotes the question and answer from the SQuAD 2.0 sample and r denotes the responses generated by the STs. $\#q$ means "number of questions". ✓$\#r$ and ✗$\#r$ denotes the number of responses which are labeled 1 and 0 respectively after the human annotation process.

Every response may be marked by zero or more annotators. When at least two annotators select the same response from the list we consider it as a *match*. To compute the annotator agreement score, we divide the number of matches with total number of annotations by each annotator. Using this formula we find average annotator agreement to be 0.665, where each annotator's agreement score is weighted by their number of annotated questions.

2.4 Evaluation of Response Classification

As previously mentioned in §2.3, the **SG** data doesn't contain all true positives since one cannot exhaustively find and annotate all the good responses when the response list is very long. Additionally, there is a large class imbalance between good and bad responses, making standard evaluation metrics such as precision, recall, F1 score and accuracy potentially misleading. To gather additional insight regarding how well the model ranks correct responses over incorrect ones, we calculate

[4] We found that some bad annotators had a high affinity of choosing the first (or the shortest) response when it was not the best choice in the list. To reduce such annotation errors we add another constraint that the shortest response should be selected by at least 2 different annotators.

Classifier	Loss	P@1	Max-F1	PR-AUC
ShortResp	-	0.324	0.189	-
LangModel	-	0.058	0.012	-
Linear	Log.	0.680	0.159	0.070
Linear	Soft.	0.640	0.387	0.344
DA	Log.	0.467	0.151	0.066
DA+ELMo	Log.	0.694	0.354	0.301
DA	Soft.	0.503	0.383	0.297
DA+ELMo	Soft.	0.716	0.456	0.427
BERT	Log.	0.816	0.490	**0.465**
BERT	Soft.	**0.833**	**0.526**	0.435

Table 2: Best response classifier results on **SG** test data. "ShortResp" stands for Shortest Response baseline, "LangModel" stands for Language Model baseline, "Linear" stands for Linear model. "Log." and "Soft." in Loss column stands for Logistic and Softmax loss respectively. DA refers to Decomposable Attention model (Parikh et al., 2016). "+ELMo" refers to adding pre-trained ELMo embeddings to DA model.

Precision@1 (P@1),[5] Max. F1,[6] and Area Under the Precision-Recall Curve (PR-AUC). We train all classifier models on the **SG** training set and evaluate them on **SG** test data. The resulting evaluation is presented in Table 2.

The results show that the shortest response baseline (ShortResp) performs worse than the ML models (0.14 to 0.51 absolute P@1 difference depending on the model). This verifies that annotation is not dominated by presentation bias where annotators are just selecting the shortest (first in the list) response for each question. The language model baseline (LangModel) performs even worse (0.41 to 0.78 absolute difference), demonstrating that this task is unlikely to have a trivial solution. The feature-based linear model shows good performance when trained with **Softmax** loss beating many of the neural models in terms of PR-AUC and Max-F1. By inspecting the weight vector, we find that grammar features, specifically the number of prepositions, determiners, and "to"s in the response, are the features with the highest weights. This probably implies that the most important challenge in this task is finding the right prepositions and determiners in the response. Other important features are the response length and the response's 3-gram LM probabilities. The ostensible limitation of feature-based models is failing to recognize correct pronouns for unfamiliar named entities in the questions.

Due to the small size of **SG** train set, the vanilla Decomposable Attention (**DA**) model is unable to learn good representations on its own and accordingly, performs worse than the linear feature-based model. The addition of ELMo embeddings appears to help to cope with this. We find that the **DA** model with ELMo embeddings is better able to predict the right pronouns for the named entities, presumably due to pre-trained representations. The best neural model in terms of P@1 is the **BERT** model fine-tuned with the **Softmax** loss (last row of Table 2).

3 Data-Augmentation and Generation

SEQ2SEQ models are very effective in generation tasks. However, our 2028 labeled question and response pairs from the **SG** train set (Table 1) are insufficient for training these large neural models. On the other hand, creating a new large-scale dataset that supports fluent answer generation by crowdsourcing is inefficient and expensive. Therefore, we augment SQuAD 2.0 with responses from the STs+**BERT** classifier (Table 2) to create a synthetic training dataset for SEQ2SEQ models. We take all the QA pairs from the SQuAD 2.0 *train-set* which can be handled by the question parser and STs, and rank their candidate responses using the **BERT** response classifier probabilities trained with **Softmax** loss (i.e. ranking loss (Collins and Koo, 2005)). Therefore, for each question we select the top ranked responses[7] by setting a threshold on the probabilities obtained from the **BERT** model. We refer to the resulting dataset as SQuAD-Synthetic (**SS**) consisting of 59,738 $\langle q, a, r \rangle$ instances.

To increase the size of **SS** training data, we take the QA pairs from Natural Questions (Kwiatkowski et al., 2019) and HarvestingQA[8] (Du and Cardie, 2018) and add $\langle q, a, r \rangle$ instances using the same STs+**BERT** classifier technique. These new pairs combined with **SS** result in a dataset of 1,051,938 $\langle q, a, r \rangle$ instances, referred to as the **SS+** dataset.

3.1 PGN, D-GPT, Variants and Baselines

Using the resulting **SS** and **SS+** datasets, we train Pointer generator networks (PGN) (See et al., 2017), DialoGPT (D-GPT) (Zhang et al., 2019) and their variants to produce a fluent answer response

[5] P@1 is the % of times the correct response is ranked first

[6] Max. F1 is the maximum F1 the model can achieve by choosing the optimal threshold in the PR curve

[7] at most three responses per question

[8] HarvestingQA is a QA dataset containing 1M QA pairs generated over 10,000 top-ranking Wikipedia articles. This dataset is noisy as the questions are automatically generated using an LSTM based encoder-decoder model (which makes use of coreference information) and the answers are extracted using a candidate answer extraction module.

generator. The input to the generation model is the question and the answer phrase $\langle q, a \rangle$ and the response r is the corresponding generation target.

PGN: PGNs are widely used SEQ2SEQ models equipped with a copy-attention mechanism capable of copying any word from the input directly into the generated output, making them well equipped to handle rare words and named entities present in questions and answer phrases. We train a 2-layer stacked bi-LSTM PGN using the OpenNMT toolkit (Klein et al., 2017) on the **SS** and **SS+** data. We additionally explore PGNs with pre-training information by initializing the embedding layer with GloVe vectors (Pennington et al., 2014) and pre-training it with $\langle q, r \rangle$ pairs from the questions-only subset of the OpenSubtitles corpus[9] (Tiedemann, 2009). This corpus contains about 14M question-response pairs in the training set and 10K pairs in the validation set. We name the pre-trained PGN model as PGN-Pre. We also fine-tune PGN-Pre on the **SS** and **SS+** data to generate two additional variants.

D-GPT: DialoGPT (i.e. dialogue generative pre-trained transformer) (Zhang et al., 2019) is a recently released large tunable automatic conversation model trained on 147M Reddit conversation-like exchanges using the GPT-2 model architecture (Radford et al., 2019). We fine-tune D-GPT on our task using the **SS** and **SS+** datasets. For comparison we also train GPT-2 on our datasets from scratch (i.e. without any pre-training). Finally, to assess the impact of pre-training datasets, we pre-train the GPT-2 on the 14M questions from questions-only subset of the OpenSubtitles data (similar to the PGN-Pre model) to get GPT-2-Pre model. The GPT-2-Pre is later fine-tuned on the **SS** and **SS+** datasets to get two corresponding variants.

CoQA Baseline: **Co**nversational **Q**uestion **A**nswering (CoQA) (Reddy et al., 2019) is a large-scale ConvQA dataset aimed at creating models which can answer the questions posed in a conversational setting. Since we are generating conversational responses for QA systems, it is sensible to compare against such ConvQA systems. We pick one of the best performing BERT-based CoQA model from the SMRCToolkit (Wu et al., 2019) as a baseline.[10] We refer to this model as the **CoQA** baseline.

QuAC Baseline: **Qu**estion **A**nswering in **C**ontext is another ConvQA dataset. We use the modified version of BiDAF model presented in (Choi et al., 2018) as a second baseline. Instead of a SEQ2SEQ generation, it selects spans from passage which acts as responses. We use the version of this model implemented in AllenNLP (Gardner et al., 2017) and refer to this model as the **QuAC** baseline.

STs+BERT Baseline: We also compare our generation models with the technique that created the **SS** and **SS+** training datasets (i.e. the responses generated by STs ranked with the **BERT** response classifier).

We validate all the SEQ2SEQ models on the human annotated **SG** data (Table 1).

3.2 Evaluation on the SQuAD 2.0 Dev Set

To have a fair and unbiased comparison, we create a new 500 question sample from the SQuAD 2.0 *dev* set (*SQuAD-dev-test*) which is unseen for all the models and baselines. This sample contains $\sim 20\%$ of the questions that cannot be handled by the STs (parser errors). For such questions, we default to outputting the answer-phrase as the response for the **STs+BERT** baseline. For the **CoQA** baseline and the **QuAC** baseline, we run their models on passages (corresponding to the questions) from *SQuAD-dev-test* to get their responses.

To demonstrate that our models too can operate in a fully automated setting like the **CoQA** baseline and the **QuAC** baseline, we generate their responses using the answer spans selected by a BERT-based SQuAD model (instead of the gold answer span from the *SQuAD-dev-test*).

For automatic evaluation we compute validation perplexity of all SEQ2SEQ generation models on **SG** data (3^{rd} column in Table 3). However, validation perplexity is a weak evaluator of generation models. Also, due to the lack of human-generated references in *SQuAD-dev-test*, we cannot use other typical generation based automatic metrics. Therefore, we use Amazon Mechanical Turk to do human evaluation. Each response is judged by 5 annotators. We ask the annotators to identify if the response is conversational and answers the question correctly. While outputting answer-phrase to all questions is trivially correct, this style of response generation seems robotic and unnatural in a prolonged conversation. Therefore, we also ask the annotators to judge if the response is a complete-sentence (e.g. "it is in Indiana") and not a sentence-fragment (e.g. "Indiana"). For each question and response pair, we show the annotators five options

[9] http://forum.opennmt.net/t/english-chatbot-model-with-opennmt/184

[10] one of the top performing model with available code.

Model	Data	PPL	a	b	c	d	e	
			✗	✓	✗	✓	✓	correct answer
			✗	✗	✓	✓	✓	complete-sentence
			-	-	-	✗	✓	grammaticality
CoQA B.	-	-	13.80	82.20	1.20	0.60	2.20	
QuAC B.	-	-	5.20	3.80	46.40	2.80	41.80	
STs+BERT B.	-	-	0.00	18.20	0.20	13.80	67.80	
PGN	SS	6.60	1.00	7.00	9.00	16.20	66.80	
PGN	SS+	3.83	1.00	3.00	8.40	17.60	70.00	
PGN-Pre	SS	4.34	0.20	4.60	9.80	17.40	68.00	
PGN-Pre	SS+	3.31	0.40	4.80	9.00	16.20	69.60	
GPT-2	SS	4.69	1.00	5.00	13.20	18.60	62.20	
GPT-2	SS+	2.70	0.80	4.20	8.20	16.80	70.00	
GPT-2-Pre	SS	3.23	0.40	2.80	8.20	19.00	69.60	
GPT-2-Pre	SS+	2.74	0.80	2.40	7.80	17.00	72.00	
D-GPT	SS	2.20	0.40	2.40	8.60	13.00	75.60	
D-GPT	SS+	2.06	0.40	2.60	7.80	13.20	**76.00**	
D-GPT (o)	SS+	2.06	0.00	3.00	0.00	13.80	**83.20**	

Table 3: Human evaluation results of all the models and baselines on sample of *SQuAD-dev-test*. In the first three rows B. stands for baseline. In the last row "(o)" stands for oracle. In Column 3 PPL stands for validation perplexity. All the values are percentage (out of 100) of responses from each model that belong to specific option(a to e) selected by annotators.

based on the three properties (correctness, grammaticality, and complete-sentence). These five options (a to e) are shown in the Table 3 header. The best response is a complete-sentence which is grammatical and answers the question correctly (i.e. option e). Other options give us more insights into different models' behavior. For each response, we assign the majority option selected by the annotators and aggregate their judgments into buckets. We present this evaluation in Table 3.

We compute the inter-annotator agreement by calculating Cohen's kappa (Cohen, 1960) between individual annotator's assignments and the aggregated majority options. The average Cohen's kappa (weighted by the number of annotations for every annotator) is 0.736 (i.e. substantial agreement).

The results reveal that **CoQA** baseline does the worst in terms of option e. The main reason for that is because most of the responses generated from this baseline are exact answer spans. Therefore, we observe that it does very well in option b (i.e. correct answer but not a complete-sentence). The **QuAC** baseline can correctly select span-based informative response $\sim 42\%$ of the time. Other times, however, it often selects a span from the passage which is related to the topic but doesn't contain the correct answer i.e. (option c). Another problem with this baseline is that it is restricted by the input passage and many not always be able to find a valid span that answers the questions. Our **STs+BERT** baseline does better in terms of option e compared to the other baselines but it is limited by the STs and the parser errors. As mentioned earlier, $\sim 20\%$ of the time this baseline directly copies the answer-phrase in the response which explains the high percentage of option b.

Almost all models perform better when trained with **SS+** data showing that the additional data from Natural Questions and HarvestingQA is helping. Except for the PGN model trained on **SS** data, all other variants perform better than **STs+BERT** baseline in terms of option e. The GPT-2 model trained on **SS** data from scratch does not perform very well because of the small size of training data. The pretraining with OpenSubtitiles questions boosts its performance (option e % for GPT-2-Pre model variants $>$ option e % for GPT-2 model variants). The best model however is D-GPT when finetuned with **SS+** dataset. While retaining the correct answer, it makes less grammatical errors (lower % in option c and d compared to other models). Furthermore with oracle answers it performs even better (last row in Table 3). This shows that D-GPT can generate better quality responses with accurate answers. We provide some sample responses from different models in Appendix A.

3.3 Evaluation on CoQA

In this section, we test our model's ability to generate conversational answers on the CoQA dev set, using **CoQA** baseline's predicted answers. The CoQA dataset consists of passages from seven different domains (out of which one is Wikipedia) and conversational questions and answers on those

Model	a	b	c	d	e
CoQA B.	12.0	78.0	5.0	2.0	3.0
D-GPT	2.0	5.0	16.0	20.0	**57.0**
D-GPT (o)	0.0	7.0	0.0	16.0	**77.0**

Table 4: Human evaluation results of **D-GPT** model (trained on **SS+** dataset) vs **CoQA** model on sample of 100 question answers from filtered CoQA dev set. (o) stands for oracle answers. Options a to e are explained in Table 3 header.

passages. Due to the conversational nature of this dataset, some of the questions are one word ($\sim 3.1\%$), like "what?", "why?" etc. Such questions are out-of-domain for our models as they require the entire context over multiple turns of the conversation to develop their response. Other out-of-domain questions include unanswerable ($\sim 0.8\%$) and yes/no ($\sim 18.4\%$) questions. We also don't consider questions with answers > 5 words ($\sim 11.6\%$) as they are typically non-factoid questions. We take a random sample of 100 from the remaining questions. This sample contains questions from a diverse set of domains outside of the Wikipedia (on which our models are trained). This includes questions taken from the middle of a conversation (for example, "who did they meet ?") which are unfamiliar for our models. We perform a human evaluation similar to §3.2 on this sample. We compare **CoQA** against **D-GPT** trained on the **SS+** dataset (with **CoQA**'s predictions input as answer-phrases). The results are shown in Table 4.

This evaluation reveals that the **D-GPT** model is able to successfully convert the **CoQA** answer spans into conversational responses 57% of the time (option e). **D-GPT** gets the wrong answer 18% of the time (option a and c), because the input answer predicted by the **CoQA** baseline is also incorrect 17% of the time. However with oracle answers, it is able to generate correct responses 77% of the times (option e). The weighted average Cohen's kappa (Cohen, 1960) score for all annotators in this evaluation is 0.750 (substantial agreement). This result demonstrates ability of our model to generalize over different domains and generate good conversational responses for questions when provided with correct answer spans.

4 Related Work

Question Generation (QG) is a well studied problem in the NLP community with many machine learning based solutions (Rus et al., 2010; Heilman and Smith, 2010; Yao et al., 2012; Labutov et al., 2015; Serban et al., 2016; Reddy et al., 2017; Du et al., 2017; Du and Cardie, 2017, 2018). In comparison, our work explores the opposite direction, i.e. (generating conversational humanlike answers given a question). Fu and Feng (2018) also try to solve fluent answer response generation task but in a restricted setting of movie related questions with 115 question patterns. In contrast, our generation models can deal with human generated questions from any domain.

Learning to Rank formulations for answer selection in QA systems is common practice, most frequently relying on *pointwise* ranking models (Severyn and Moschitti, 2015; Garg et al., 2019). Our use of discriminative re-ranking (Collins and Koo, 2005) with softmax loss is closer to learning a *pairwise* ranking by maximizing the multiclass margin between correct and incorrect answers (Joachims, 2002; Burges et al., 2005; Köppel et al., 2019). This is an important distinction from TREC-style answer selection as our ST-generated candidate responses have lower semantic, syntactic, and lexical variance, making pointwise methods less effective.

Question Answering Using crowd-sourcing methods to create QA datasets (Rajpurkar et al., 2016; Bajaj et al., 2016; Rajpurkar et al., 2018), conversational datasets (Dinan et al., 2018), and ConvQA datasets (Choi et al., 2018; Reddy et al., 2019; Elgohary et al., 2018; Saha et al., 2018) has largely driven recent methodological advances. However, models trained on these ConvQA datasets typically select exact answer spans instead of generating them (Yatskar, 2019). Instead of creating another crowd-sourced dataset for our task, we augment existing QA datasets to include such conversational answer responses using the STs + BERT trained with softmax loss.

5 Conclusion

In this work, we study the problem of generating fluent QA responses in the context of building fluent conversational agents. To this end, we propose an over-generate and rank data augmentation procedure based on Syntactic Transformations and a best response classifier. This method is used to modify the SQuAD 2.0 dataset such that it includes conversational answers, which is used to train SEQ2SEQ based generation models. Human evaluations on *SQuAD-dev-test* show that our models generate

significantly better conversational responses compared to the baseline CoQA and QuAC models. Furthermore, the D-GPT model with oracle answers is able to generate conversational responses on the CoQA dev set 77 % of the time showcasing the model's scalability.

Acknowledgments

We would like to thank the reviewers for providing valuble feedback on an earlier draft of this paper. This material is based in part on research sponsored by the NSF (IIS-1845670), ODNI and IARPA via the BETTER program (2019-19051600004) DARPA via the ARO (W911NF-17-C-0095) in addition to an Amazon Research Award. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of NSF, ODNI, ARO, IARPA, DARPA or the U.S. Government.

References

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

Christopher Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Gregory N Hullender. 2005. Learning to rank using gradient descent. In *Proceedings of the 22nd International Conference on Machine learning (ICML-05)*, pages 89–96.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wentau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. QuAC: Question answering in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2174–2184, Brussels, Belgium. Association for Computational Linguistics.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Michael Collins and Terry Koo. 2005. Discriminative reranking for natural language parsing. *Computational Linguistics*, 31(1):25–70.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2018. Wizard of wikipedia: Knowledge-powered conversational agents. *arXiv preprint arXiv:1811.01241*.

Xinya Du and Claire Cardie. 2017. Identifying where to focus in reading comprehension for neural question generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2067–2073, Copenhagen, Denmark. Association for Computational Linguistics.

Xinya Du and Claire Cardie. 2018. Harvesting paragraph-level question-answer pairs from Wikipedia. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1907–1917, Melbourne, Australia. Association for Computational Linguistics.

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1342–1352, Vancouver, Canada. Association for Computational Linguistics.

Ahmed Elgohary, Chen Zhao, and Jordan Boyd-Graber. 2018. A dataset and baselines for sequential open-domain question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1077–1083, Brussels, Belgium. Association for Computational Linguistics.

Yao Fu and Yansong Feng. 2018. Natural answer generation with heterogeneous memory. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 185–195, New Orleans, Louisiana. Association for Computational Linguistics.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke S. Zettlemoyer. 2017. Allennlp: A deep semantic natural language processing platform.

Siddhant Garg, Thuy Vu, and Alessandro Moschitti. 2019. Tanda: Transfer and adapt pre-trained transformer models for answer sentence selection. *arXiv preprint arXiv:1911.04118*.

Albert Gatt and Ehud Reiter. 2009. Simplenlg: A realisation engine for practical applications. In *Proceedings of the 12th European Workshop on Natural Language Generation (ENLG 2009)*, pages 90–93.

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

Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142. ACM.

John Judge, Aoife Cahill, and Josef Van Genabith. 2006. Questionbank: Creating a corpus of parse-annotated questions. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 497–504. Association for Computational Linguistics.

Dan Klein and Christopher D Manning. 2003. Fast exact inference with a factored model for natural language parsing. In *Advances in neural information processing systems*, pages 3–10.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proc. ACL*.

Marius Köppel, Alexander Segner, Martin Wagener, Lukas Pensel, Andreas Karwath, and Stefan Kramer. 2019. Pairwise learning to rank by neural networks revisited: Reconstruction, theoretical analysis and practical performance. *arXiv preprint arXiv:1909.02768*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Igor Labutov, Sumit Basu, and Lucy Vanderwende. 2015. Deep questions without deep understanding. 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 889–898, Beijing, China. Association for Computational Linguistics.

Roger Levy and Galen Andrew. 2006. Tregex and tsurgeon: tools for querying and manipulating tree data structures. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2249–2255, Austin, Texas. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Sathish Reddy, Dinesh Raghu, Mitesh M. Khapra, and Sachindra Joshi. 2017. Generating natural language question-answer pairs from a knowledge graph using a RNN based question generation model. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 376–385, Valencia, Spain. Association for Computational Linguistics.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. CoQA: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Vasile Rus, Brendan Wyse, Paul Piwek, Mihai Lintean, Svetlana Stoyanchev, and Cristian Moldovan. 2010. The first question generation shared task evaluation challenge.

Amrita Saha, Vardaan Pahuja, Mitesh M Khapra, Karthik Sankaranarayanan, and Sarath Chandar. 2018. Complex sequential question answering: Towards learning to converse over linked question answer pairs with a knowledge graph. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Iulian Vlad Serban, Alberto García-Durán, Caglar Gulcehre, Sungjin Ahn, Sarath Chandar, Aaron Courville, and Yoshua Bengio. 2016. Generating factoid questions with recurrent neural networks: The 30M factoid question-answer corpus. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 588–598, Berlin, Germany. Association for Computational Linguistics.

Aliaksei Severyn and Alessandro Moschitti. 2015. Learning to rank short text pairs with convolutional deep neural networks. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pages 373–382. ACM.

Jörg Tiedemann. 2009. News from opus-a collection of multilingual parallel corpora with tools and interfaces. In *Recent advances in natural language processing*, volume 5, pages 237–248.

Stephen Wan, Mark Dras, Robert Dale, and Cécile Paris. 2006. Using dependency-based features to take the'para-farce'out of paraphrase. In *Proceedings of the Australasian Language Technology Workshop 2006*, pages 131–138.

Jindou Wu, Yunlun Yang, Chao Deng, Hongyi Tang, Bingning Wang, Haoze Sun, Ting Yao, and Qi Zhang. 2019. Sogou Machine Reading Comprehension Toolkit. *arXiv e-prints*, page arXiv:1903.11848.

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

Mark Yatskar. 2019. A qualitative comparison of CoQA, SQuAD 2.0 and QuAC. 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 2318–2323, Minneapolis, Minnesota. Association for Computational Linguistics.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2019. Dialogpt: Large-scale generative pre-training for conversational response generation.

A Sample responses from different models

Sample of responses from different models on *SQuAD-dev-test* set §3.2. "**Q**:" is the Question, "**R**:" is the Response, "B." stands for baseline and "(o)" stands for oracle answer spans.					
Model	**Q**:what fixed set of factors determine the actions of a deterministic turing machine	correctness	complete-sentence	grammaticality	majority option
CoQA B.	**R**: fixed set of rules	✓	✗	-	b
QuAC B.	**R**: a deterministic turing machine is the most basic turing machine , which uses a fixed set of rules to determine its future actions	✓	✓	✓	e
STs+BERT B.	**R**: rules determine the actions	✓	✓	✓	e
PGN-Pre with SS+	**R**: fixed set of rules determine the actions	✓	✓	✓	e
GPT-2 with SS+	**R**: fixed set of rules fixed set of factors determine the actions	✓	✓	✗	d
GPT-2-Pre with SS+	**R**: fixed set of rules determine the actions	✓	✓	✓	e
D-GPT with SS+	**R**: fixed set of rules determine the actions	✓	✓	✓	e
D-GPT with SS+ (o)	**R**: rules determine the actions	✓	✓	✓	e
Model	**Q**:why was polonia relegated from the country 's top flight in 2013 ?	correctness	complete-sentence	grammaticality	majority option
CoQA B.	**R**: because of their disastrous financial situation	✓	✗	-	b
QuAC B.	**R**: polonia was relegated from the country 's top flight in 2013 because of their disastrous financial situation	✓	✓	✓	e
LGRs+BERT B.	**R**: polonia was relegated from the country 's top flight because the disastrous financial situation	✓	✓	✗	d
PGN-Pre with SS+	**R**: polonia was relegated disastrous financial situation	✓	✓	✗	d
GPT-2 with SS+	**R**: polonia was relegated disastrous financial situation	✓	✓	✗	d
GPT-2-Pre with SS+	**R**: polonia was relegated disastrous financial situation	✓	✓	✗	d
D-GPT with SS+	**R**: polonia was relegated from the country's top flight for disastrous financial situation	✓	✓	✓	e
D-GPT with SS+ (o)	**R**: polonia was relegated from the country's top flight for disastrous financial situation	✓	✓	✓	e
Model	**Q**:how was sadat rewarded by the islamists for his attempts to bring egypt into modern times and civilization ?	correctness	complete-sentence	grammaticality	majority option
CoQA B.	**R**: released Islamists from prison	✗	✗	-	a
QuAC B.	**R**: egyptian president anwar sadat – whose policies included opening egypt to western investment (infitah) ; transferring egypt 's allegiance from the soviet union to the united states	✗	✓	-	c
LGRs+BERT B.	**R**: sadat was rewarded by the islamists for his attempts to bring egypt assassinated	✓	✓	✗	d
PGN-Pre with SS+	**R**: sadat was rewarded piety and charity	✗	✓	-	c
GPT-2 with SS+	**R**: it was rewarded piety and charity	✗	✓	-	c
GPT-2-Pre with SS+	**R**: sadat was rewarded by piety and charity	✗	✓	-	c
D-GPT with SS+	**R**: sadat was rewarded piety and charity	✗	✓	-	c
D-GPT with SS+ (o)	**R**: sadat was assassinated	✓	✓	✓	e

Model	**Q**:from what type of materials must liquid oxygen be separated ?	correctness	complete-sentence	grammaticality	majority option
CoQA B.	**R**: combustible materials	✓	✗	-	b
QuAC B.	**R**: both liquid and solid o n2 are clear substances with a light sky-blue color caused by absorption in the red	✗	✓	-	c
STs+BERT B.	**R**: combustible materials	✓	✗	-	b
PGN-Pre with SS+	**R**: it must be separated from combustible	✓	✓	✓	e
GPT-2 with SS+	**R**: it must be separated from combustible	✓	✓	✓	e
GPT-2-Pre with SS+	**R**: they must be separated from combustible	✓	✓	✓	e
D-GPT with SS+	**R**: liquid oxygen must be separated from combustible	✓	✓	✓	e
D-GPT with SS+ (o)	**R**: liquid oxygen must be separated from combustible materials	✓	✓	✓	e
Model	**Q**:on what type of exponentiation does the diffie – hellman key exchange depend on ?	correctness	complete-sentence	grammaticality	majority option
CoQA B.	**R**: modular exponentiation	✓	✗	-	b
QuAC B.	**R**: several public-key cryptography algorithms , such as rsa and the diffie – hellman key exchange	✗	✗	-	a
STs+BERT B.	**R**: modular exponentiation	✓	✗	-	b
PGN-Pre with SS+	**R**: it depends on modular	✓	✓	✓	e
GPT-2 with SS+	**R**: it depends on modular	✓	✓	✓	e
GPT-2-Pre with SS+	**R**: it depends on the modular	✓	✓	✓	e
D-GPT with SS+	**R**: it depends on modular	✓	✓	✓	e
D-GPT with SS+ (o)	**R**: it depends on modular exponentiation	✓	✓	✓	e
Model	**Q**:what was sadat seeking by releasing islamists from prison ?	correctness	complete-sentence	grammaticality	majority option
CoQA B.	**R**: political support in his struggle against leftists	✓	✓	✓	e
QuAC B.	**R**: egyptian president anwar sadat – whose policies included opening egypt to western investment (infitah) ; transferring egypt 's allegiance from the soviet union to the united states	✗	✓	-	c
STs+BERT B.	**R**: sadat was seeking political support	✓	✓	✓	e
PGN-Pre with SS+	**R**: it was making peace with israel	✗	✓	-	c
GPT-2 with SS+	**R**: it was making peace with israel	✗	✓	-	c
GPT-2-Pre with SS+	**R**: sadat seeking was making peace with israel	✗	✓	-	c
D-GPT with SS+	**R**: sadat was making peace with israel	✗	✓	-	c
D-GPT with SS+ (o)	**R**: sadat was seeking political support	✓	✓	✓	e
Model	**Q**:how did the better jacksonville plan generate money ?	correctness	complete-sentence	grammaticality	majority option
CoQA B.	**R**: authorized a half-penny sales tax	✓	✗	-	b
QuAC B.	**R**: when a consolidation referendum was held in 1967 , voters approved the plan . on october 1 , 1968 , the governments merged to create the consolidated city of jacksonville	✗	✓	-	c
STs+BERT B.	**R**: it generated money authorized a half-penny sales tax	✓	✓	✗	d
PGN-Pre with SS+	**R**: it generated money half-penny sales tax	✓	✓	✗	d

GPT-2 with SS+	**R**: it generated money half-penny sales tax	✓	✓	✗	d
GPT-2-Pre with SS+	**R**: it generated half-penny sales tax money	✓	✓	✓	e
D-GPT with SS+	**R**: it generated money for half-penny sales tax	✓	✓	✓	e
D-GPT with SS+ (o)	**R**: it generated money authorized a half-penny sales tax	✓	✓	✗	d
Model	**Q**:how many items is the university library home to ?	correctness	complete-sentence	grammaticality	majority option
CoQA B.	**R**: over two million	✓	✗	-	b
QuAC B.	**R**: another important library – the university library , founded in 1816 , is home to over two million items	✓	✓	✓	e
STs+BERT B.	**R**: it to is over two million	✓	✓	✗	d
PGN-Pre with SS+	**R**: it is home to over two million	✓	✓	✓	e
GPT-2 with SS+	**R**: it is home to over two million	✓	✓	✓	e
GPT-2-Pre with SS+	**R**: it is home to over two million	✓	✓	✓	e
D-GPT with SS+	**R**: it to is over two million	✓	✓	✗	d
D-GPT with SS+ (o)	**R**: it to is over two million	✓	✓	✗	d
Model	**Q**:which sea was oil discovered in ?	correctness	complete-sentence	grammaticality	majority option
CoQA B.	**R**: North Sea	✓	✗	-	b
QuAC B.	**R**: ” it ’ s scotland ’ s oil ” campaign of the scottish national party (snp)	✗	✗	-	a
LGRs+BERT B.	**R**: oil was discovered in north	✓	✓	✗	d
PGN-Pre with SS+	**R**: oil was discovered in the north sea	✓	✓	✓	e
GPT-2 with SS+	**R**: oil was discovered in the north sea	✓	✓	✓	e
GPT-2-Pre with SS+	**R**: it was discovered in the north sea	✓	✓	✓	e
D-GPT with SS+	**R**: it was discovered in the north sea	✓	✓	✓	e
D-GPT with SS+ (o)	**R**: oil was discovered in north	✓	✓	✗	d
Model	**Q**:where are jersey and guernsey	correctness	complete-sentence	grammaticality	majority option
CoQA B.	**R**: Channel Islands	✓	✗	-	b
QuAC B.	**R**: the customary law of normandy was developed between the 10th and 13th centuries and survives today through the legal systems of jersey and guernsey in the channel islands	✓	✓	✓	e
LGRs+BERT B.	**R**: they are in channel islands	✓	✓	✓	e
PGN-Pre with SS+	**R**: they are in the channel islands	✓	✓	✓	e
GPT-2 with SS+	**R**: they are on the channel islands	✓	✓	✓	e
GPT-2-Pre with SS+	**R**: they are on the channel islands	✓	✓	✓	e
D-GPT with SS+	**R**: they are in the channel islands	✓	✓	✓	e
D-GPT with SS+ (o)	**R**: they are in channel islands	✓	✓	✓	e
Model	**Q**:near chur , which direction does the rhine turn ?	correctness	complete-sentence	grammaticality	majority option
CoQA B.	**R**: north	✓	✗	-	b

QuAC B.	**R**: near tamins-reichenau the anterior rhine and the posterior rhine join and form the rhine	✗	✓	-	c
LGRs+BERT B.	**R**: it turns north	✓	✓	✓	e
PGN-Pre with SS+	**R**: it turns north	✓	✓	✓	e
GPT-2 with SS+	**R**: it turns north	✓	✓	✓	e
GPT-2-Pre with SS+	**R**: it turns to the north	✓	✓	✓	e
D-GPT with SS+	**R**: it turns north	✓	✓	✓	e
D-GPT with SS+ (o)	**R**: it turns north	✓	✓	✓	e
Model	**Q**:what kind of contract is given when the contractor is given a performance specification and must undertake the project from design to construction , while adhering to the performance specifications ?	correctness	complete-sentence	grammaticality	majority option
CoQA B.	**R**: design build" contract	✓	✗	-	b
QuAC B.	**R**: the modern trend in design is toward integration of previously separated specialties , especially among large firms	✗	✓	-	c
LGRs+BERT B.	**R**: a "design build" contract is given	✓	✓	✓	e
PGN-Pre with SS+	**R**: design build is given	✓	✓	✓	e
GPT-2 with SS+	**R**: the design build is given	✓	✓	✓	e
GPT-2-Pre with SS+	**R**: design build is given a performance specification and must undertake the project	✓	✓	✓	e
D-GPT with SS+	**R**: design build is given	✓	✓	✓	e
D-GPT with SS+ (o)	**R**: the " design build " contract is given	✓	✓	✓	e
Model	**Q**:how many protestants live in france today ?	correctness	complete-sentence	grammaticality	majority option
CoQA B.	**R**: Approximately one million	✓	✗	-	b
QuAC B.	**R**: approximately one million protestants in modern france represent some 2 % of its population	✓	✓	✓	e
LGRs+BERT B.	**R**: one million live in france today	✓	✓	✓	e
PGN-Pre with SS+	**R**: one million live in france today	✓	✓	✓	e
GPT-2 with SS+	**R**: one million live in france today	✓	✓	✓	e
GPT-2-Pre with SS+	**R**: one million live in france today	✓	✓	✓	e
D-GPT with SS+	**R**: one million live in france today	✓	✓	✓	e
D-GPT with SS+ (o)	**R**: one million live in france today	✓	✓	✓	e
Model	**Q**:what is raghuram rajan 's career ?	correctness	complete-sentence	grammaticality	majority option
CoQA B.	**R**: Central Banking economist	✓	✗	-	b
QuAC B.	**R**: central banking economist raghuram rajan argues that " systematic economic inequalities	✓	✗	-	b
LGRs+BERT B.	**R**: he is economist	✓	✓	✗	d
PGN-Pre with SS+	**R**: it is central banking economist	✓	✓	✓	e
GPT-2 with SS+	**R**: it is central banking economist	✓	✓	✓	e
GPT-2-Pre with SS+	**R**: it is central banking economist	✓	✓	✓	e
D-GPT with SS+	**R**: it is central banking economist	✓	✓	✓	e
D-GPT with SS+ (o)	**R**: he is economist	✓	✓	✗	d

Model	**Q**:what type of steam engines produced most power up to the early 20th century ?	correctness	complete-sentence	grammaticality	majority option
CoQA B.	**R**: Reciprocating piston type steam engines	✓	✗	-	b
QuAC B.	**R**: reciprocating piston type steam engines remained the dominant source of power until the early 20th century , when advances in the design of electric motors and internal combustion engines	✓	✓	✓	e
LGRs+BERT B.	**R**: reciprocating piston produced most power up	✓	✓	✗	d
PGN-Pre with SS+	**R**: reciprocating piston type produced most power up	✓	✓	✗	d
GPT-2 with SS+	**R**: reciprocating piston type produced most power up	✓	✓	✗	d
GPT-2-Pre with SS+	**R**: the reciprocating piston type produced most power up to the early 20th century	✓	✓	✓	e
D-GPT with SS+	**R**: reciprocating piston type produced most power up to the early 20th century	✓	✓	✓	e
D-GPT with SS+ (o)	**R**: reciprocating piston produced most power up to the early 20th century	✓	✓	✓	e
Model	**Q**:where did france win a war in the 1950 's	correctness	complete-sentence	grammaticality	majority option
CoQA B.	**R**: Algeria	✓	✗	-	b
QuAC B.	**R**: france fought and lost a bitter war in vietnam in the 1950s	✗	✓	-	c
LGRs+BERT B.	**R**: france won a war in the 1950 's algeria	✓	✓	✓	e
PGN-Pre with SS+	**R**: france won a war in vietnam	✗	✓	-	c
GPT-2 with SS+	**R**: france won a war in vietnam	✗	✓	-	c
GPT-2-Pre with SS+	**R**: france won a war in vietnam	✗	✓	-	c
D-GPT with SS+	**R**: france won a war in vietnam	✗	✓	-	c
D-GPT with SS+ (o)	**R**: france won a war in algeria	✓	✓	✓	e
Model	**Q**:who did the ottoman empire ally with in ww i ?	correctness	complete-sentence	grammaticality	majority option
CoQA B.	**R**: Germany	✓	✗	-	b
QuAC B.	**R**: the ottoman empire gradually declined into the late nineteenth century . the empire allied with germany	✓	✓	✓	e
LGRs+BERT B.	**R**: germany did the ottoman empire ally with in ww i	✓	✓	✗	d
PGN-Pre with SS+	**R**: it separated with germany	✗	✓	-	c
GPT-2 with SS+	**R**: it allyed with germany	✓	✓	✓	e
GPT-2-Pre with SS+	**R**: it allyed with germany	✓	✓	✓	e
D-GPT with SS+	**R**: it allied germany	✓	✓	✗	d
D-GPT with SS+ (o)	**R**: it allied germany	✓	✓	✗	d
Model	**Q**:when was ambulatory care pharmacy approved as its own certification ?	correctness	complete-sentence	grammaticality	majority option
CoQA B.	**R**: In 2011	✓	✗	-	b
QuAC B.	**R**: in 2011 the board of pharmaceutical specialties approved ambulatory care pharmacy practice as a separate board certification	✓	✓	✓	e
LGRs+BERT B.	**R**: it was approved in 2011	✓	✓	✓	e

PGN-Pre with SS+	**R**: it was approved in 2011	✓	✓	✓	e
GPT-2 with SS+	**R**: it was approved in 2011	✓	✓	✓	e
GPT-2-Pre with SS+	**R**: it was approved in 2011	✓	✓	✓	e
D-GPT with SS+	**R**: it was approved in 2011	✓	✓	✓	e
D-GPT with SS+ (o)	**R**: it was approved in 2011	✓	✓	✓	e
Model	**Q**:when did arpnet and sita become operational	correctness	complete-sentence	grammaticality	majority option
CoQA B.	**R**: 1969	✓	✗	-	b
QuAC B.	**R**: arpanet and sita hln became operational in 1969	✓	✓	✓	e
LGRs+BERT B.	**R**: 1969	✓	✗	-	b
PGN-Pre with SS+	**R**: they became operational in 1969	✓	✓	✓	e
GPT-2 with SS+	**R**: they became operational in 1969	✓	✓	✓	e
GPT-2-Pre with SS+	**R**: they became operational in 1969	✓	✓	✓	e
D-GPT with SS+	**R**: they became operational in 1969	✓	✓	✓	e
D-GPT with SS+ (o)	**R**: they became operational in 1969	✓	✓	✓	e
Model	**Q**:how much did saudi arabia spend on spreading wahhabism ?	correctness	complete-sentence	grammaticality	majority option
CoQA B.	**R**: over 100 billion dollars	✓	✗	-	b
QuAC B.	**R**: saudi arabia spent over 100 billion dollars in the ensuing decades for helping spread its fundamentalist interpretation of islam	✓	✓	✓	e
LGRs+BERT B.	**R**: saudi arabia spent over 100 billion dollars	✓	✓	✓	e
PGN-Pre with SS+	**R**: saudi arabia spent over 100 billion dollars	✓	✓	✓	e
GPT-2 with SS+	**R**: saudi arabia spent over 100 billion dollars	✓	✓	✓	e
GPT-2-Pre with SS+	**R**: saudi arabia spent over 100 billion dollars	✓	✓	✓	e
D-GPT with SS+	**R**: saudi arabia spent over 100 billion dollars	✓	✓	✓	e
D-GPT with SS+ (o)	**R**: saudi arabia spent over 100 billion dollars	✓	✓	✓	e