Natural Answer Generation: From Factoid Answer to Full-length Answer using Grammar Correction

Anonymous ACL submission

Abstract

Question Answering systems these days typically use template-based language generation. Though adequate for a domain-specific task, these systems are too restrictive and predefined for domain-independent systems. This paper proposes a system that outputs a full-length answer given a question and the extracted factoid answer (short spans such as named entities) as the input. Our system uses constituency and dependency parse trees of questions. A transformer-based Grammar Error Correction model GECToR is used as a post-processing step for better fluency. We compare our system with (i) a Modified Pointer Generator (SOTA) and (ii) Fine-tuned DialoGPT for factoid questions. We also tested our approach on existential (yes-no) questions with better results. Our model generates more accurate and fluent answers than the state-of-the-art (SOTA) approaches. The evaluation is done on NewsQA and SqUAD datasets with an increment of 0.4 and 0.9 percentage points in ROUGE-1 score respectively. Also, the inference time is reduced by 85% compared to the SOTA. The improved datasets used for our evaluation will be released as part of the research contribution.

1 Introduction

Question answering (QA) is an exercise of finding solutions for a query from a given paragraph. Normally small spans of text, inclusive of named entities, dates, etc. are extracted as answers. However, knowledge-base (KB) orientated QA systems extract factoid solutions by using a structured query or neural representation of the question. As a natural extension and post-processing step, the retrieved factoid answer is transformed into a full-length natural sentence. Unlike conversational chat-bots designed to mimic human communique without worrying to be factually correct, or assignment-orientated dialogue system which places the retrieved solution in a predefined template, our approach routinely generates correct full-length solutions, thereby, improving its utilization in these situations.

Question *: Who was the duke in the battle of hastings ?*
Factoid answer *: William the conqueror*
Target *: [The duke in the battle of hastings was William the conqueror. , William the conqueror was the duke in the battle of hastings.]*

Example 1 - Sample from SqUAD dataset

Our overall research contributions are listed as follows:

1. We achieve superior performance by incorporating a pre-trained transformer encoder GEC sequence tagging system as a post-processing step in our rule-based approach. In our experiments, encoders from RoBERTa outperform three other cutting-edge transformer encoders (XLNet, BERT).

2. We present a rule-based approach for existential questions (Yes/No questions) where Yes/No is considered as the factoid answer and the natural answer is generated by rearranging noun phrases and verb phrases present in the question. We achieve good metrics (BLEU, ROUGE-1,2,L) and also analyze the results of using the Grammar correction model, GECTOR, on top of the developed rule-based system.

3. We have made the existing dataset for this task more accurate by correcting grammar errors in GOLD answers and have added alternate answers wherever necessary. We also have created a small dataset for Existential QA having different types of indirect questions as well. We will open-source all the improved datasets for further research.

The rest of the paper is organized as follows: Firstly we discuss some recent works and related literature in section 2, after which we give details about the data used for evaluating our system in section 3. After that we talk about our approach in section 4; rule based in section 4.1 (factoid questions in section 4.1.1, existential questions in section 4.1.2) and fine-tuned DialoGPT in section 4.2. Following up on this, we provide details about our experimental setup and discuss the GCM used as a post-processing step in section 5. Then in section 6 we provide the results & evaluation of our approach; compare performance from other approaches. Then in section 7, we give extensive error analysis of all approaches (Modified Pointer Generator [SOTA] in section 7.1, fine-tuned DialoGPT in section 7.2 and rule-based in section 7.3) presented in the paper and discuss some ways to overcome them. Lastly, we conclude our paper by discussing future work in section 8.

2 Related Work

Recently there has been a lot of work in question answering and dialog systems; most of this work in question answering has been extracting answer span in the context paragraph, often referred to as machine comprehension or extracting answer nodes in a knowledge graph (KB-based Question Answering). Here in this paper, we present the task of natural answer generation or generating fluent responses given a question and its factoid answer. There are very few models or papers which deal with this end-to-end problem response generation problem where after extracting the short answer span, do not generate the human-like full-length answer. But due to the increase in information on the web, extracting relevant information presently is a critical task. This is increasingly becoming time consuming task as well because of the increase in online data. This has prompted the development of various robust question answering systems or information extraction widely used today in search engines. These systems though robust and accurate in finding the relevant information return the short answer to the question asked, not a fluent full-length answer which has high application in various user-centric chatbots and voice assistants.
In most of the recent works, we have observed a question answering system where the answer is in the form of paragraphs (more than 1 sentence) extracted from online retrieved passages. (Asai et al., 2018), (Du and Cardie, 2018), (Wang et al., 2017), (Wang and Jiang, 2016), (Oh et al., 2016), or span-based exact answer from a reading comprehension or knowledge base. (Chen et al., 2017).
On the contrary, the task of natural answer generation has received little attention. There has been some work indirectly related to this task (Brill et al., 2002), which was done to maximize the answer patterns(retrieved documents) by reordering the words of the question.
Some recent works are presented in (Pal et al., 2019) and (Akermi et al., 2020). Former work tried to tackle this issue by proposing a supervised approach based on modifying pointer generator network (See et al., 2017) while the latter proposed a transformer-based unsupervised approach incorporating language models to evaluate different possible answer structures. In (Pal et al., 2019), the model was trained on a novel dataset made from multiple existing machine comprehension datasets with manual annotations, this end-to-end neural supervised approach didn't generalize well and was not accurate in many cases. In (Akermi et al., 2020), authors have used a syntactic parser to form rules to get fragments useful for forming natural answers. They assumed that only one word could be missing and it should be located before the factoid answer within the identified structure. This assumption cannot be generalized and can lead to incomplete answers with grammatical errors.
Our answer generation approach differs from these works as it is entirely rule-based. The rules we have used can be generalized because of the use of a syntactic parse tree of the question, which is the most effective way of forming rules. We have utilized (Omelianchuk et al., 2020) by which any number of words at any place can be added or deleted. Indeed, we build upon the intuitive hypothesis that a full length can be made by reformulating the words given in the question and factoid answer with few insertions/deletions in between, which we are handling using a transformer-based grammar error correction model.

3 Data

There is just one available dataset (Pal et al., 2019) for this task created from a reading comprehension dataset having 15000 manually annotated, 300000 automatically annotated from SQuAD (Rajpurkar et al., 2016), HarvertingQA (Du and Cardie, 2018), and 420 data points in test dataset taken from

NewsQA (Trischler et al., 2017). After going through the dataset, we realized the available dataset is not of high quality, having multiple grammatically incorrect questions/answers and also wrong or grammatically incorrect target answers in many cases. Due to this, improving the quality of the dataset is the need of the hour.
In natural language generation (NLG) systems, there can be more than one correct answer that is not incorporated well in the available dataset.

Question *: Who is the CEO of google ?*
Factoid answer *: Sundar Pichai*
Target *: [(i) Sundar Pichai is the CEO of google. (ii) The CEO of google is Sundar Pichai.]*

In the existing dataset, we see only target (i) type annotations but target (ii) is also the correct way to answer this question and should be added to the annotation. So we improve the quality of the available dataset to handle the above-mentioned issues. We sampled 7200 data points from 15000 manually annotated SqUAD samples (Pal et al., 2019), 420 data points from NewsQA (Pal et al., 2019) and made the required changes in target answers; some data points were removed due to incomplete question/answer. As given in Table 1, our improved dataset has 6768 data points from SQuAD and 380 data points from NewsQA. We have also created 166 data points of the existential QA dataset containing different varieties and forms of asking questions, including indirect questions. The codes and the data sets will be publicly available after the acceptance of the paper.

Question - type (i) *: Does my fridge support quick freeze feature?*
Question - type (ii) *: Can you tell me if my fridge supports quick freeze feature?*
Target *: [No, your fridge does not support quick freeze feature. OR Yes, your fridge supports quick freeze feature.]*

Example 2 - Sample from Yes/No dataset

4 Approach

In this section we explain the rule based approach and fine-tuned DialoGPT approach developed.

Dataset	Count
NewsQA (Factoid)	380
SqUAD (Factoid)	6768
Yes/No (Existential)	166

Table 1: Dataset used for our evaluation

4.1 Rule Based Approach

4.1.1 Factoid Questions

After observing a large number of examples in the available dataset we were able to find patterns in the formation of the natural answers using the sentence structure of the question at its core. Initially, the idea was to check the accuracy by just replacing the *WH* words present in the question with the factoid answer; we refer to that approach as Rule Based V1 in the below examples. Analyzing the output of the above idea on the failed cases led to a finding of patterns related to the position of the auxiliary verb and the main verb. We used the constituency and dependency parsing output of the question to find positions of auxiliary verbs, main verbs, noun phrases, and verb phrases present in the question and designed the algorithm; we refer to this improved version of our approach as Rule Based V2 (RBV2). Outputs of constituency parser with Elmo Embeddings given in (Joshi et al., 2018) and deep biaffine attention neural dependency parser (Dozat and Manning, 2017) were extensively used in the algorithm developed. We used open source AllenNLP library (Gardner et al., 2017) APIs of the above 2 parsers in developing our rule based system.
Below we will explain our approach using some examples and also discuss implementation details. In the first version of our rule based approach (Rule Based V1), we have just replaced the *WH* words (*what, when, why, who, how etc.*) present in the question with the factoid answer. The *WH* word in the question was found by using the outputs of POS tags of the AllenNLP constituency parser (Joshi et al., 2018). If the tag is "WP" or "WRB" or "WDT" then we replace that word with a factoid answer. This phenomenon where the sentence topic appears at the front of the sentence as opposed to in a canonical position further to the right is known as topicalization (Prince, 1998). Some examples are stated below for a better understanding of the approach:-

Question *: What is the capital of India?*
Factoid answer *: Delhi*
Rule Based V1 *: Delhi is the capital of India*
Target answer *: Delhi is the capital of India*

Example 3 - Self made Sample

Question *: what was the space station crew forced to take shelter from?*
Factoid answer *: a piece of debris*
Rule Based V1 *: a piece of debris was the space station crew forced to take shelter from*
Target answer *: the space station crew was forced to take shelter from a piece of debris*

Example 4 - Sample from NewsQA dataset

In the second version (Rule Based V2[RBV2]), we modify the above approach based on the position of the auxiliary verb and main verb present in the question. We formulate the algorithm to solve the problem of the ordering of natural answers, i.e., answer followed by portion from a question or portion of a question followed by the answer. So, if the main verb and auxiliary verb are consecutive, the factoid answer appears in the starting otherwise we add it at the end. In the latter case, we start our answer from the word after the auxiliary verb, till the main verb is encountered, then the auxiliary word is added to the answer string. Then we copy the part of the question after the main verb, finally adding the factoid answer.
If the question does not have a verb in it then we add all words after the auxiliary word present in the question to our answer, then add the auxiliary verb, and finally add the factoid answer. Some sample example outputs using this approach are stated below:-

Question *: What is the capital of India?*
Factoid answer *: Delhi*
Rule Based V2(RBV2) *: the capital of India is Delhi*

CASE :- Main Verb not present

Question *: what was the space station crew forced to take shelter from?*
Factoid answer *: a piece of debris*
Rule Based V2(RBV2 *: the space station crew was forced to take shelter from a piece of debris*

CASE :- Auxiliary Verb and Main Verb not together

4.1.2 Existential Questions (Yes/No)

It would be incomplete if we limit this task of natural answering to just factoid questions. This task can have importance in the existential question type and in systems or apps tackling user queries using speech assistants or chatbots. So, we tried formulating a rule based approach for existential or yes/no questions using the dependency and constituency parse tree of the questions. Generally, such questions have a common structure: auxiliary verb (AUX) followed by a noun phrase (NP) and then a verb phrase (VP) in the end, i.e., AUX-NP-VP. The natural answers to such questions can be made by reordering the above parts to NP-AUX-VP. This was implemented using the output of the AllenNLP dependency parse tree model. In addition, we start the answer with "yes," or "no," so as to create a more natural-sounding answer.

Question *: Can you tell if fridge supports quick freeze feature?*
Factoid answer *: Yes*
RB *: Yes, fridge does supports quick freeze feature.*
RB + RoBERTa *:Yes, fridge does support quick freeze feature.*

Example 5 - Sample from Yes/No dataset

4.2 Fine-tuned DailoGPT

In order to resolve the problem of fluency which is very important for the task of generating natural human-like full-length answers, we used autoregressive language models which generate human-like fluent text. Amongst all the autoregressive LMs we selected GPT2 (Radford et al., 2019) model because of its large size of training data and number of parameters. This 1.5B transformer model achieved state-of-the-art results on most language modeling datasets on zero short learning tasks. For our task, we needed a neural conversational response generation model, finding for some existing work in the conversational dialog systems using autoregressive LMs like GPT2, we found the DialoGPT (DGPT) (Zhang et al., 2020) model. DGPT model is a conversational dialog system or chatbot and produces very fluent human-like text taking the most recent text as input and

the previous conversations as context to generate the response. DGPT is an extension of the GPT2 model trained on 147M conversations from Reddit. As it was claimed in the paper (Zhang et al., 2020) that conversational agents leveraging the DGPT model were producing human-like fluent text and the model was able to generate responses that were consistent with the context and relevant to the recent prompt/question/chat. It was also shown in the paper that DGPT generated responses were very much similar to humans by performing extensive human evaluation and also through automatic evaluation using various metrics. Also, since all the datasets used, the training pipeline, pretrained model was open-sourced by the authors, which made using this model and performing experiments very less time taking. This made using DGPT model our first choice amongst all the other models because of the similarity in our task of human like response generation to questions, and the task DGPT was trained. The only difference in both these tasks was in the context part, in our task the context was the short answer span (factoid answer). Also in our task, the data we used consisted of independent question answers pairs, different from the Reddit comment chains training data used in DGPT which may have subsequent questions of the related context as the previous ones. We believed that if DGPT generates responses that are coherent, and relevant to the context, then it is worth analyzing its performance in our setting. Hence, we fine-tuned the above pre-trained model on approximately 13000 manually annotated questions, short answer, and full length answer triplets given by (Pal et al., 2019).

Typically DialoGPT model was created to make conversational chatbots, and their fine-tuning is also done for building conversational agents where the input is the question asked, and all the previous dialogues are kept as a series of contexts and are passed as input to the model for training. This has applications in making conversational chatbots relevant to a particular field. For instance, suppose a chatbot that has the knowledge of a particular book, movie, etc is required, then all the dialogs exchanges can be used to train the DGPT model, and then all the responses from the trained model will have all the required context. But here, for our task, we concatenate the question with its extracted factoid answer and keep manually annotated answers as targets in fine-tuning the model. For inference, question and factoid answers are concatenated and provided as input to the fine-tuned model to generate a response.

5 Experimental Setup

We have used Tesla T4 16GB GPU to carry out the experiments. For factoid questions, we have used two datasets having 380 and 6768 data points as given in Table 1. Experimental results are shown in Table 2 and 3, respectively. For existential questions, we have used created data set with 166 examples. Results of confirmatory dataset are reported in Table 4.

As a post processing step of all our rule based approaches, we have used a pre-trained transformer encoder, grammar error correction (GEC) given in (Omelianchuk et al., 2020). This model was available with 3 cutting edge transformer encoders, namely BERT, RoBERTa, and XLNet. Experiments were carried out using all 3 above encoder based GEC models as post processing steps in our rule based approach.

For fine-tuning DialoGPT, we took a pretrained DialoGPT-small (117M parameters) and fine-tuned with around 13000 manually annotated samples data from (Pal et al., 2019). We trained the model for 8 epochs. The results on 380 data points (cross-validation) of NewsQA dataset by the fine-tuned model are reported in Table 2.

6 Results

We use standard BLEU (Papineni et al., 2002) (NLTK), ROUGE-1, 2, L (Lin, 2004) (rouge-score) metrics to evaluate our system and compare our system with other approaches. In Table 2, 3, 4 : "RBV2+RoBERTa" means our rule based approach with grammar correction performed by RoBERTa encoder and so on.

Table 5 illustrates a qualitative comparison of outputs from different approaches explored in this paper.

In Table 2, we see an increase in BLEU, ROUGE-2, ROUGE-L scores on using RoBERTa encoder Grammar Correction Model (GCM) as compared to not using it. It is also clear that RoBERTa based encoder GCM is superior as compared to other encoders due to higher BLEU and ROUGE scores. Our developed approach attains very comparable results in terms of BLEU and ROUGE-1, 2, L scores and reduces inference time by 85% as compared to the state of the art MPG model. Avg.

time in table 2, 3 denotes the average time taken by the model or algorithm to generate an answer for 1 (question, factoid answer) input. ROUGE-1 and ROUGE-L scores are almost the same with a difference of 3 and 1 percentage points in BLEU and ROUGE-2 scores, respectively. BLEU and ROUGE scores provided in all the tables are on a scale of 100.

In Table 3, reported ROUGE-1 and ROUGE-L scores are almost the same. BLEU and ROUGE-2 scores for our approach (RBV2 + GCM) are a bit lesser than the SOTA model (MPG).

There are instances in the above tables were employing a GCM sometimes reduces the BLEU or ROUGE scores, especially in Table 3. This phenomenon is very much related to the target (GOLD) answers based on which the scores are calculated. This can occur because of insertion/deletion of punctuation in between by GCM but not present in the target answer and vice-versa. In many cases, target answers do not follow correct grammar which sometimes leads to lower scores. But in such cases also the overall quality, fluency, and adequacy of the answers improved by GCM are much better.

Table 2 illustrates that the performance of fine-tuned DialoGPT is comparatively very low as compared to other approaches in cross evaluation. The main problem with this approach was the problem of hallucination as explained in (Maynez et al., 2020) which decreases the accuracy of the approach, and hence we conclude that it is not useful for this task. Due to that, we have skipped the results of the fine-Tuned model in Table 3.

In Table 4, scores are calculated on a very small dataset and the best scores are achieved by simply employing the rule based model without using GCM. We still argue to use of a GCM as a post processing step in this type as well due to its ability to improve the overall quality of the answers. This improvement in quality can not be measured using these scores but can surely improve user satisfaction. This kind of task in existential questions is to the best of our knowledge first time presented so there is no baseline model to compare our results with.

7 Error Analysis

Below we present some qualitative discussion and error analysis of answers generated by existing approaches and our proposed approach.

7.1 Modified Pointer Generator(MPG)

This approach was taken from (Pal et al., 2019). The main limitations of this approach are stated in the below points. Also, there were failure cases wherein the model just outputs the question itself which may be due to the model becoming biased towards adding more parts from the question than the factoid answer which results in complete copying of the question in some cases. Below are the main types of failure cases stated:-

- Incoherent sentence due to failure in reasoning
- Repetition of words
- Outputs only the factoid answer
- Outputs clausal answers
- Failure to incorporate morphological variations

This can also be seen in Table 5 where MPG makes errors in answer generation. Word positions of were and going are interchanged and "at" is added which is wrong, the correct addition should be "to". Overall, this model doesn't attain good results even for very straightforward example cases present in our dataset and so using it for general case queries would not be very beneficial. Also, the inference time of this model is very high (last column of Table 2,3).

7.2 Fine-tuned DialoGPT

The problem of adding unwanted things in the final answers which don't have any mention in the question and the factoid answer often called hallucination (Maynez et al., 2020) is the main shortcoming of this model.

There are instances where a factoid answer is not even present in the final answer. Also, there are numerous cases where the DialoGPT model makes errors in copying numerical data for *e.g.* year, number, etc. The model has some errors in copying the proper nouns as given in the questions. The final answer has those names but with changed spelling. (*e.g.*:- elizabeth - elizabetha; alexander - alexandrick). This is also evident from the example given in Table 5 where DialoGPT has changed arizona spelling to "anrizona". This leads to low BLEU and ROUGE scores. For eg,

Question *: What is going live on tuesday?*
Factoid answer *: web-based on-demand television*

Model	BLEU	ROUGE-1	ROUGE-2	ROUGE-L	Avg. time (sec.)
MPG(2019)	84.9	95.7	89.4	93.9	2.54
RBV2	79.1	96.1	85.5	93.1	0.382
RBV2+BERT	77.6	94.4	85.4	92.4	0.397
RBV2+RoBERTa	81.7	95.7	88.2	93.6	0.394
RBV2+XLNET	80.3	94.8	87.0	92.9	0.4
DialoGPT	50.3	73.4	49.3	70.0	0.908

Table 2: Results on 380 data points of NewsQA dataset

Model	BLEU	ROUGE-1	ROUGE-2	ROUGE-L	Avg. time (sec.)
MPG(2019)	75.8	94.4	87.4	91.6	2.54
RBV2	74.8	95.3	83.1	90.3	0.399
RBV2+BERT	71.5	93.9	82.4	89.5	0.411
RBV2+RoBERTa	72.1	94.0	83.1	89.8	0.411
RBV2+XLNET	71.2	93.6	82.3	89.4	0.413

Table 3: Results on 6768 data points of SqUAD dataset

Model	BLEU	R-1	R-2	R-L
RB	70.2	87.3	75.0	84.8
RB+BERT	62.7	85.5	71.6	83.4
RB+RoBERTa	66.6	84.5	73.0	84.2
RB+XLNET	67.5	86.6	74.0	84.6

Table 4: Results on 166 data points of existential questions dataset created by us; Here in the table R represents Rouge, R-1 means ROUGE-1 and so on

and movie service
***Fine-Tuned DialoGPT** : on tuesday, the web-based version of "net based" television and film service.*
***Target answer** : web-based on-demand television and movie service is going live on tuesday.*

Example 6 - Sample from NewsQA dataset

In the above example we find very poor quality of answer generated. Here we see additional "net-based" getting added which makes this model unreliable for this task.

7.3 Rule Based Model

This approach works by reordering question sentence structure and copy pasting the factoid answer, and so if the factoid answer is not factual based or is a clausal answer then this approach may fail. Also, the generated answers may be grammatically wrong in terms of missing a word like in, is, to etc. which is corrected by the transformer based grammar correction used as a post processing step; other types of grammatical error by rule based approach is incorrect positioning of AUX word (*e.g.* is, are, etc) in the answer which is not corrected by the (Omelianchuk et al., 2020) sometimes.

***Question** : where did lewis partnership begin?*
***Factoid answer** : started as a single shop on oxford street in london, opened in 1864 by john.*
***RBV2** : lewis partnership begin started as a single shop on oxford street in london, opened in 1864 by john.*
***Target answer** : lewis partnership begin started as a single shop on oxford street in london, opened in 1864 by john.*

Example 7 - Sample from SqUAD dataset

In the above example, the output answer had both begin and started in it which is not right, this is because the factoid answer contains a clause having a verb part included. Currently, in our system, we are not checking the factoid answer structure to define our answers, and hence for these examples, this model may fail. Since the approach works on the question structure so if the question is not properly well-formed or incomplete then the answers will not be correct. In instances where the question is of type "how many"; the word "many" can be added or not added based on the type of factoid answer given. In such cases, we rely on the GCM model to perform necessary corrections

Input	Output
Ques - where was the bus going ?	**MPG** (Pal et al., 2019) - the bus going was at phoenix, arizona.
Factoid Ans. - phoenix, arizona	**FT DialoGPT** [ours] - the bus was going to phoenix, anrizona.
	RBV2 [ours] - the bus was going phoenix, arizona .
	RBV2+GCM [ours] - The bus was going to Phoenix, Arizona.

Table 5: Comparison of outputs from all approaches discussed in the paper for an input example. Here MPG represents the state of the art deep learning model using the Pointer Generator technique. FT DialoGPT represents the results of the fine-tuned model of DialoGPT for this task. RBV2+GCM represents the results of using the GEC Model as a post processing step. Here we used the RoBERTa encoder GECTOR model as GCM.

Grammar Error	Count
Grammar Error [extra]	103
Grammar Error [incorrect]	25
Grammar Error [misplaced]	254
Grammar Error [missing]	815

Table 6: Count of categories of grammar errors by the rule based algorithm without using the GCM. These numbers are for the 6768 data points from SqUAD dataset

but sometimes the GCM model fails to make the changes.

Questions having a subordinate clause are a challenge to this approach. Such examples generally have 2 WH words and so sometimes are difficult to handle. With some modifications, we will be able to handle those questions as well in our rule-based approach by first finding out the main clause in the question and masking the subordinate clause temporarily considering if that subordinate clause never existed, and then unmasking it after answer generation.

As highlighted by van Miltenburg et al. (2021), the under-reporting of errors and lack of extensive error analysis of NLG system output is quite common nowadays. This prevents researchers to get an idea about the specific weakness of SOTA and the improved model. So in this work, we categorized the errors for the 6768 data points of the SQuAD dataset. These errors are categorized as extra words like do, does, is, was; incorrect words like much, many; misplaced words like is, were, was, are, has; missing words like in, to, on, during, by, until, through, at, after, between; wrong preposition, word order. The count of these categories is reported in 6. The GCM as the post-processing step in our approach is able to correct most of the above errors for our system and thus improve the quality of our generated answers as can be seen in Table 5.

8 Conclusion & Future Work

In this work, we have worked on the task of generating full-length natural answers given the question and the factoid answer. We have solved this task by designing a rule based approach using the syntactic parser. A Grammar Correction Model (GCM) is used as a post processing step to improve the fluency of generated natural answer. Our approach RBV2 and RoBERTa based encoder GCM achieves superior results than the state of art deep learning model in terms of ROUGE-1 score, quality of the answers generated, and inference time. This system can be used at the final stage of any domain-specific QA system or answering user troubleshooting queries where factoid answer is extracted by a knowledge base or context paragraphs. This approach is developed using general rules of answer generation and so can be applied to all domains as compared to a supervised system which gets biased to the type of training data given. We have also improved the quality of the existing dataset by creating 2 sets having 6768 and 380 data points, respectively. We have also created a dataset of 166 data points of existential (yes/no) questions.

We plan to make our system more robust, especially for questions having subordinate clauses present. We will work on making a complete system that can classify existential and factoid questions and use our developed system on top of that. We plan to give our generated answers for review to some proficient English speakers and ask for scores on fluency, adequacy of our generated answer, and other approaches' answers. Further work needs to be done to investigate the performance of reinforcement learning based techniques for solving this task, keeping BLEU or ROUGE score as the reward. We plan on adding more variation to the data by annotating and correcting additional QA pairs both in factoid and existential questions.

References

Imen Akermi, Johannes Heinecke, and Frédéric Herledan. 2020. Tansformer based natural language generation for question-answering. In *Proceedings of the 13th International Conference on Natural Language Generation, INLG 2020, Dublin, Ireland, December 15-18, 2020*, pages 349–359. Association for Computational Linguistics.

Akari Asai, Akiko Eriguchi, Kazuma Hashimoto, and Yoshimasa Tsuruoka. 2018. Multilingual extractive reading comprehension by runtime machine translation. *CoRR*, abs/1809.03275.

Eric Brill, Susan Dumais, and Michele Banko. 2002. An analysis of the AskMSR question-answering system. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 257–264. Association for Computational Linguistics.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. *ArXiv*, abs/1611.01734.

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.

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.

V. Joshi, Matthew E. Peters, and Mark Hopkins. 2018. Extending a parser to distant domains using a few dozen partially annotated examples. In *ACL*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

Jong-Hoon Oh, Kentaro Torisawa, Chikara Hashimoto, Ryu Iida, Masahiro Tanaka, and Julien Kloetzer. 2016. A semi-supervised learning approach to why-question answering. *Proceedings of the AAAI Conference on Artificial Intelligence*, 30(1).

Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanskyi. 2020. GECToR – grammatical error correction: Tag, not rewrite. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 163–170, Seattle, WA, USA â†' Online. Association for Computational Linguistics.

Vaishali Pal, Manish Shrivastava, and Irshad Bhat. 2019. Answering naturally: Factoid to full length answer generation. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 1–9, Hong Kong, China. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

E. Prince. 1998. On the limits of syntax, with reference to left-dislocation and topicalization.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

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.

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.

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

Emiel van Miltenburg, Miruna Clinciu, Ondřej Dušek, Dimitra Gkatzia, Stephanie Inglis, Leo Leppänen, Saad Mahamood, Emma Manning, Stephanie Schoch, Craig Thomson, and Luou Wen. 2021. Underreporting of errors in NLG output, and what to do about it. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 140–153, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Shuohang Wang and Jing Jiang. 2016. Machine comprehension using match-lstm and answer pointer. *CoRR*, abs/1608.07905.

Tong Wang, Xingdi Yuan, and Adam Trischler. 2017. A joint model for question answering and question generation. *CoRR*, abs/1706.01450.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. DIALOGPT : Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278, Online. Association for Computational Linguistics.