Natural Answer Generation: Factoid Answer to Full Length Answer

BTP-I & II Report

By

Manas Jain

Roll No. 170040068

Under the guidance of

Prof. Pushpak Bhattacharyya, Prof. Jayadipta Ghosh

Department of Civil Engineering
Indian Institute of Technology, Bombay
2020-21

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Date: 16$^{\text{th}}$ June 2021	Manas Jain
Place: IIT Bombay, Mumbai	Roll No: 170040068

Acknowledgements

I would like to thank **Prof. Pushpak Bhattacharyya and Prof. Jayadipta Ghosh** for their keen guidance and constant support. All the meetings and discussions were highly useful in understanding the problem and exploring the possible solutions to the problem. I would also like to thank **Prof. Sriparna Saha** for her constant guidance and support throughout the project. I am highly indebted to all LG-Soft Researchers such as **Mr. Gladvin Chinnadurai** and **Mr. Manish Vatsa** for their guidance and constant supervision as well as for providing necessary information regarding the project. I would also like to thank all members of CFILT lab for their help and support.

Abstract

In Question Answering domain, to present the user with a more conversational experience the task of generation of ”full length answer” from factoid answer becomes very important. In recent years, the task of Question Answering over passages (reading comprehension) has evolved into a very active research area. A reading comprehension system extracts a span of text, consisting of named entities etc., which serve as the answer to a given question (known as ”factoid answer”). However, these spans of text would result in an unnatural reading experience to user in a systems like chatbots and speech assistants. Usually, dialogue systems solve this issue by using template-based language generation. These systems, though adequate for a domain-specific task, are too restrictive and predefined for a domain-independent system. Our system outputs a full-length answer given as input a question and the extracted factoid answer. In this work we present a full length natural answer generation system using an unsupervised rule-based approach built on using constituency and dependency parse tree of question with transformer based grammar correction model GECTOR (2020) as a post processing step. Comparison was made with 2 supervised approaches (i) the Modified Pointer Generator (SOTA) (ii) Fine-tuned DialoGPT. The developed system shows very promising results and improvement in rouge scores and inference time over state of the art model.

The project is in collaboration with LG-Soft. The project is about creating a response template to a user query where the factoid answer is extracted from product manuals using span based Q/A techniques. In this report, the development of these approaches to solve the task of natural answer generation is described in detail. We have compared the results and models, analysed reasons why a particular approach is not giving good results with examples.

Contents

List of Figures vii

List of Tables viii

1 Introduction 1

1.1 Motivation . . . 2

1.2 Problem Statement . . . 2

1.3 Approach And Research Plan . . . 3

1.3.1 Scope . . . 4

1.4 Contribution . . . 4

1.5 Organization of Report . . . 4

1.6 Summary . . . 5

2 Literature Survey 6

2.1 Introduction of Recent Summarization and Machine Translation techniques used in Neural Natural Answer Generation . . . 7

2.1.1 Basic Neural Machine Translation (NMT) . . . 8

2.1.2 Neural Attention model of summarization . . . 9

2.2 Pointer-Generator Network . . . 9

2.2.1 Baseline Sequence-to-Sequence Attention model . . . 10

2.2.2 Pointer-Generator Network . . . 11

2.3 Transformer based Natural Language Generation for Question Answering (Akermi et al. (2020)) . . . 12

2.4 Summary . . . 13

3 Dataset Description 14

3.1 Open Domain Dataset . . . 14

3.1.1 Auto Annotation Technique . . . 15

3.1.2 Manually Annotated Data . . . 15

3.2 LG-Soft Dataset . . . 16

3.3 Dataset Creation . . . 16

3.4 Summary . . . 17

4 Modified Pointer Generator Approach 18

4.1 Architecture . . . 18

4.2 Results . . . 20

4.3 Limitations and Error Analysis . . . 21

4.4 Summary . . . 23

5 Finetuned DialoGPT-2 Approach 24

5.1 Brief introduction of DialoGPT model . . . 24

5.2 Architecture of DIALOGPT model . . . 25

5.3 Experiments . . . 26

5.4 Results and model Output . . . 26

5.5 Model Limitations and Error Analysis . . . 27

5.6 Summary . . . 29

6 Rule Based Approach 30

6.1 Ideation . . . 30

6.2 Approach . . . 31

6.3 Results and Model Output . . . 32

6.4 Model Limitations and Error Analysis . . . 33

6.5 Summary . . . 34

7 Post Processing Step : Grammar Correction Model 35

7.1 Introduction . . . 35

7.2 Experiments and Results . . . 36

7.3 Qualitative Analysis . . . 38

7.4 Error Analysis . . . 39

7.5 Summary . . . 40

8 Conclusion and Future Work 41

8.1 Summary 41

8.2 Conclusion 42

8.3 Future Work 43

8.3.1 Deep Learning based models (Bilstm, transformers) . . 43

8.3.2 Rule Based Model 44

List of Figures

2.1 Baseline Sequence-to-Sequence Attention model for Abstractive Summarization (See et al., 2017) 10
2.2 Pointer-Generator Model for Abstractive Text Summarization (See et al., 2017) 11

4.1 Modified Pointer Generator Architecture (Pal et al., 2019) . . 19
4.2 Results of Modified Pointer Generator Approach (Pal et al., 2019) 21
4.3 Failure cases of the Modified Pointer Generator Approach (Pal et al., 2019) 22

5.1 DGPT model sample output examples 28

6.1 Rule Based v2 model sample output examples 33

List of Tables

5.1 DGPT Model results on 420 examples from NewsQA dataset . 27

5.2 DGPT Model results on 3200 examples from manually annotated test dataset 27

6.1 Rule Based Model results on 420 examples from NewsQA dataset 33

7.1 Results on 380 examples of NewsQA dataset 37

7.2 Results on 6768 examples of SqUAD dataset 37

7.3 Results on 840 examples of Freebase dataset 38

7.4 Results on 166 examples of Confirmatory questions dataset . . 38

Chapter 1

Introduction

Factoid question answering (QA) is the task of extracting answers for a question from a given passage. These answers are usually short spans of text, such as named entities, dates, etc. Modern factoid QA systems which use machine comprehension datasets, predict the answer span from relevant documents using encoder-decoder architectures with co-attention. Conversely, knowledge-base (KB) oriented QA systems retrieve relevant facts using structured queries or neural representation of the question. Formulating the retrieved factoid answer into a full-length natural sentence is, hence, a natural extension and post-processing step of any QA system. A simple approach for this task might be to use hand-crafted rules to restructure the question into a declarative statement. However, such rule based approaches fail when the extracted answer span, contains words from the question or when there are multiple independent clauses and the system has to choose words specific to the question to formulate the answer. This leads to unnatural repetition of words in the full-length answer or grammatically incorrect sentence formulation. On the other hand, neural-network based approaches in modern dialogue systems use end-to end encoder-decoder architectures to convert an abstract dialogue action into natural language utterances. Such modern task-oriented dialogue systems usually learn to map dialogue histories to system response. Non-task oriented dialogue systems such as generative systems can formulate responses not present in the training data but lacks the capability to incorporate factual information without external knowledge bases.

1.1 Motivation

Applications like task-oriented conversational agents or chatbots often rely on QA systems to return factually correct responses to queries, but need to generate Natural Language Responses. Current QA systems usually return an answer span in the available context, or a Knowledge Base fact triplet (Subject, Predicate, Object). Using existing state-of-the art QA systems to generate full length natural responses is a natural extension of such systems. Exploration of hybrid neural approaches using abstractive & extractive techniques simultaneously and rule based systems using constituency and dependency parse of the question. Unlike conversational chat-bots designed to mimic human conversation without the need to be factually correct, or task-oriented dialogue systems which place the retrieved answer in a predefined template, our system automatically generates accurate full-length answers, thereby, enhancing the system's usage in these situations. This system can be used in any such task-specific scenarios where natural answers are desired, by a hybrid system which combines template based answer with the neural based response which are not restricted to a limited set of templates.

1.2 Problem Statement

Generate a response template (Natural answer) i.e generate a full length answer given a question and its factoid answer as input. Example :-

- Sample Input:
  - Question : When were the normans in normandy?
  - Factoid Answer : 10th and 11th centuries
- Output: Any 1 of the 2 below
  - During the **10th and 11th centuries** , the normans were in normandy.
  - The normans were in normandy during the **10th and 11th centuries**.

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.]*

1.3 Approach And Research Plan

To solve the above task firstly neural approaches were explored wherein we fed the both the input as given in the neural network and train the system to generate a sequence as a full length answer. There was a new and recent approach used to solve this task which used the technique popular in text summarization to solve the task of natural answering. The similarity between summarization and natural answering was in both cases we have to copy some things as it is from the input and add some connecting words in between. This required literature survey of current summarization tehnique and understanding the pointer generator technique in detail. Since this used both extractive and abstractive summarization techniques simultaneously hence these topics were also explored.

After the summarization approach, one completely abstractive dialogue generation approach was implemented to see if language quality and fluency can improve the natural answers of a given question.

Finally a rule based technique based on using the constituency and dependency parse of the question was implemented. The question was restructured into a declarative statement and then the position to add factoid answer was decided based on language structure of the question.

The research plan was to come with a neural approach to solve the task of natural answering which can be used to answer every kind of question. After analysing results and limitations of approaches implemented, steps to improve that algorithm was looked on before moving on to try a completely different approach.

Finally a hybrid approach combining all the models would be the best becuase with proper conditions of which model to use to output the answer the final output examples will have less number of failure examples.

1.3.1 Scope

The report will discuss the approaches implemented and their results on a question answering dataset. Also comparison between different models, limitations, error analysis of all approaches will be discussed. Example outputs of every model and its comparison with GOLD standard will also be reported.

1.4 Contribution

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) and output quality of RoBERTa model is best.

2. We also apply this approach to confirmatory type questions (Yes/No questions) where yes and no is considered as the factoid answer. We have got good results in confirmatory type questions with the rule based approach with a Grammar correction model as the post processing step.

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

1.5 Organization of Report

In Chapter 2, we describe the related research done in other places. Related approaches which were used to solve similar tasks are discussed in this chapter. Attention and Pointer Generator Summarization technique is also discussed

In Chapter 3, we have illustrated the dataset used in this project. Discussed about both the data openly available and LG-Soft data extracted from product manuals

In Chapter 4, we explain about the Modified Pointer Generator approach, results obtained and error analysis.

In chapter 5, we have described the approach of finetuning the DialoGPT2 model on the factoid answer-full length answer dataset. Discussed the results obtained with error analysis

In Chapter 6, we explain about the rule based approach based on constituency and dependency parse of question implemented. Results and error analysis of this approach are discussed

In Chapter 7, we discuss about the post processing step; using transformer based Grammar Error Correction model. We compare and analyse results of the approaches implemented with and without GEC model with examples.

In Chapter 8, we conclude the project and propose future work to complete the project.

1.6 Summary

In this chapter, we discussed why converting a factoid answer to full length answer is important. Then we discussed the problem statement of the project. We discuss about the approaches implemented to solve the task of conversion of factoid answer to a full length answer in the approach and research plan section. We then discuss about our contribution made in this project. At last, we state the overall structure of the report with chapter description in the organization of the report section .

Chapter 2

Literature Survey

There has been a lot of interest recently in QA and task-oriented dialogue systems. End-to-end memory networks use a language modelling architecture which learns query embeddings in addition to input and output memory representations from source sequences and predicts an answer. Rule based systems such as (Weston et al., 2015) sets up a variety of tasks for inferring and answering the question. Some improvement on the memory networks is based handles out-of-vocabulary (OOV) words by inserting special words into the vocabulary for each knowledge base entity types. These systems are dependent on templates or special heuristics to reproduce facts. We demonstrate through our baseline model that generating template-like sentences from factual input can be achieved with limited success. Recent works on KB-based end-to-end QA systems such as (Yin et al., 2016), (He et al., 2017), (Liu et al., 2018) generate full-length answers with neural pointer networks (Gulcehre et al., 2016) after retrieving facts from a knowledge base (KB). Dialogue systems such as (Lian et al., 2019) (Liu et al., 2018) extract information from knowledge bases to formulate a response. Systems such as (Fu and Feng, 2018) uses KB based key-value memory after extracting information from documents or external KBs. However, these systems are restricted to only information modeled by the KB or slot-value memory. Our system, is generic and can be used with any knowledge source, structured such as a knowledge base or free form such as machine-comprehension dataset. Since our system doesn't use any additional relational information as modelled in a KB, it is invariant to the type of dataset. The pointer generator network, introduced in (See et al., 2017), is a generative summarization model that can copy out-of-vocabulary (OOV) words from a source sequence. Our work is inspired from the ability of this network to accurately reproduce information from source. To the best of our knowledge, there is no existing QA data-set which addresses the task directly. However, Knowledge-based QA dataset such as (Yin et al., 2016) creates a knowledgebase from Chinese websites and extracts questionanswer pairs from Chinese communityQA webpage. The system built over this dataset, is able to generate natural answers to simple questions. The recently released CoQA dataset (Reddy et al., 2019) is an abstractive conversational question answering dataset through which the system generates free-form answers from the whole conversational history using the aforementioned pointergenerator network. While the CoQA challenge extracts free-form text from the passages, our system incorporates the structure of the question to give a full-length sentence as answer to the given query.

2.1 Introduction of Recent Summarization and Machine Translation techniques used in Neural Natural Answer Generation

Recently due to the success of neural networks, various newer approaches using deep neural networks are getting proposed. Along with summarization, Machine Translation(MT) is also getting discussed. In machine translation, input text presented in one language is converted to another language. Like summarization, machine translation also makes use of language interpretation and generation modules. One of the groundbreaking approaches of MT is presented in section 2.1.1 proposed by (Bahdanau et al., 2016). On top of the basic idea presented in the approach of Neural Machine Translation(NMT), various other approaches were proposed, summarization using sequence-to-sequence RNN is one of them. Recurring Neural Network(RNN) is an advanced form of feedforward neural network, where the neuron is used to train recursively in a single pass of training and such multiple passes can be used to train the model. Enhancements are made to the Sequenceto-Sequence(S2S) models by adding the notion of attention. Attention allows the model to search for a specific location to learn from. Neural Attention Model for Sentence Summarization as explained in section 2.1.2 is one of such models. Finally, we describe pointer-generator network model (See et al., 2017) in section 2.2 for text summarization, which enhances attention model by probabilistically choosing between generation and extraction.

2.1.1 Basic Neural Machine Translation (NMT)

Jointly Learning to Align and Translate is an approach to the neural machine translation, where input available in one language is translated to target language. The Natural language Generation (NLG) as can be used to perform task of translation. In this section, we describe the translation using deep neural network approach. The notion of alignment captures the mapping between word generated as part of the output and the words present in the source sentence whereas translation has usual meaning of converting from source language to target language. Traditional NMT approaches haven't captured the alignment part and they were working at phrase level, whereas (Bahdanau et al., 2016) works at sentence level meaning that at a time single sentence gets translated to the target sentence.

- **Background:** There are three basic steps in any type of encoder-decoder model. For each step, various parameters need to be learnt. Method of learning parameter is highly dependent on the objective function that the model tries to learn.
  - **Encoder State:** Model encodes input sequence in a suitable format, in case of NMT each word in input sentence is represented using fixed length vector which is also called as the embedding of the word. Embedding can be seen as a representation of word the in the continuous space. Word2Vec, Golve are two widely used embedding techniques.
  - **Hidden State:** This is a black box step, where encoded input is transformed to produce output of same length as the input.

Simply length of vector produced by hidden state is same as the length of the column of embedding matrix. Number of hidden steps varies as per the suggested model.

- **Decoder State:** Decoder step reverses the process done by the encoder step and generat word on the basis of its embedding.

2.1.2 Neural Attention model of summarization

It is also called as Attention Based Summarization(ABS) (Rush et al., 2015). It tries to make use of the linguistic structure for generation of summaries, for that it captures the attention in input sequence to produce correct output. This is an extension to the model presented in section 2.1.1 and successor of this model is presented in section 2.2. The approach presented in the paper (Rush et al., 2015) is of abstractive sentence summarization which takes sentence as input and converts it into a condensed form. This approach can be further extended to produce summary of documents. The approach proposed makes use of basic feedforward neural networks and generates probability distribution over output sequence. Encoder takes input words and already generated words as input and transforms this using feature matrix.To train the model negative log likelihood objective function is used and to generate the summary sentence. In decoder beam search algorithm is proposed, as complicity of greedy algorithm is exponential in terms of the window.

2.2 Pointer-Generator Network

Recent summarization approaches discussed so far tries to generate the summaries irrespective of correctness of factual data and without considering novelty of information in produced summary. Abstractive summarization proposed in this paper (See et al., 2017) tries to overcome these shortcomings along with handling of OOVs. The author discusses three approaches (1)Baseline model (section 2.2.1) (2)Pointer generator model (section 2.2.2)

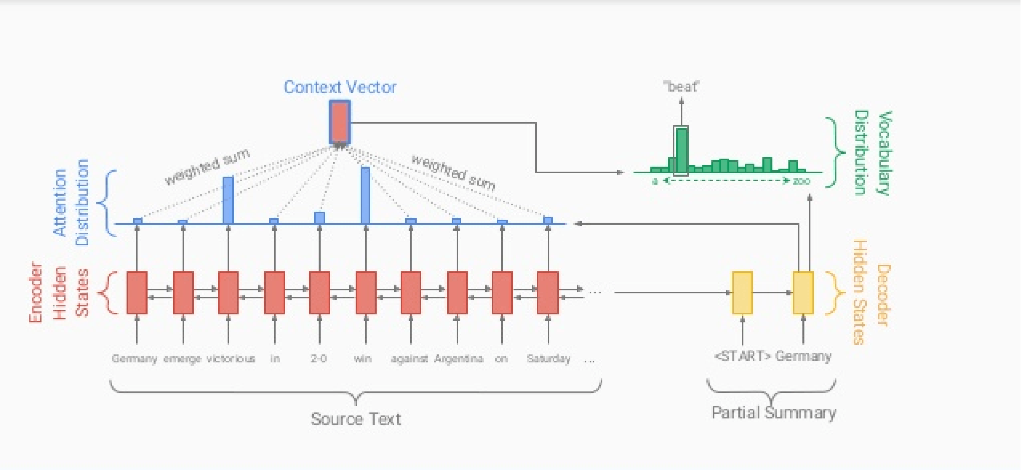


Figure 2.1: Baseline Sequence-to-Sequence Attention model for Abstractive Summarization (See et al., 2017)

2.2.1 Baseline Sequence-to-Sequence Attention model

This section discusses the baseline Sequence-to-Sequence Attention model for Abstractive Summarization . Proposed baseline uses single bidirectional LSTM as encoder and single layer unidirectional LSTM as decoder. This baseline model is depicted in figure 2.1 from which it gets clear that the word beat gets generate based on present context of sentence. Let encoder hidden states be h_i and decoder hidden states be s_i then attention distribution at time-step t can be formulated as shown in equation 2.1 and 2.2 where $v, W_h, W_s$and b_a are learnable parameters.

$$e_i^t = v^T tanh(W_h h_i + W_s s^t + b_a) \tag{2.1}$$

$$a_t = softmax(e_t) \tag{2.2}$$

Attention can be considered as the location to produce next word from. Attention is used to get weighted sum of hidden state which represents overall hidden state h^*. This hidden state along with hidden state of decoder then used to probability distribution P_v over all words in vocabulary.

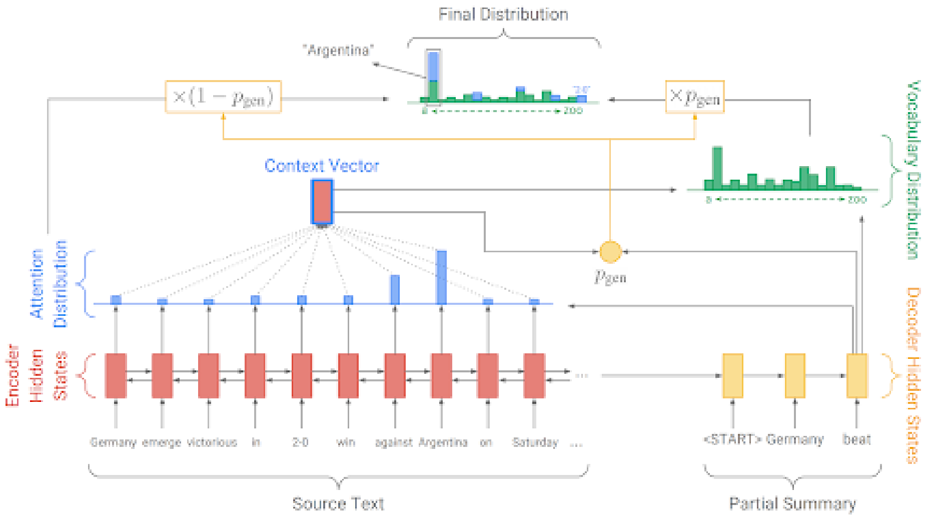


Figure 2.2: Pointer-Generator Model for Abstractive Text Summarization (See et al., 2017)

2.2.2 Pointer-Generator Network

This is hybrid model which combines the baseline model and the model of pointer network proposed in Vinyals et al., 2015. Pointer generation model tries to handle OOVs either by copying from input text or by generating from decoder vocabulary. Figure 2.2 describes use of working of pointer-generator model. The proposed model switch is modelled as continuous variable between range [0, 1]. The authors call this is a soft switch and a function is used to decide between generation and pointer mechanism. The notion of extended vocabulary which is a combination of vocabulary and all words appearing in the input text. When w happens to be OOV, P_v becomes zero and in case of non-appearance in source document, attention term becomes zero. Negative log-likelihood is used as loss function to train the model and learn the parametes.

2.3 Transformer based Natural Language Generation for Question Answering (Akermi et al. (2020))

The work presented in Pal et al. (2019) and Akermi et al. (2020) tried to tackle this task of Natural Answer Generation for QA by proposing a supervised approach and unsupervised approach respectively. In Pal et al. (2019) the model was trained on a small data set whose questions/answers pairs were extracted from machine comprehension data-sets and augmented manually which make generalization and capturing variation very limited. In Akermi et al. (2020) they have used syntactic parser to form rules to get fragments useful for the formation of natural answer. They assume 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.

The work in Akermi et al. (2020) used Cutting edge transformers Language models to solve this task. Also to predict this missing word, they use BERT as the generation model (GM) for its ability to capture bidirectionally the context of a given word within a sentence. Their assumption was that one word could be missing and that it is located before the short answer within the identified structure, as it could be the case for a missing article (the, a, etc.) or a preposition (in, at, etc.) for example.

The following example illustrates the different steps of the approach proposed in Akermi et al. (2020):

Question: When did princess Diana die?

1. Question parsing and answer extraction using state of art machine comprehension system: short answer = August 31, 1997
2. Chunking the question into text fragments using the UDPipe based dependency analysis: Q=When, did die, princess Diana
3. Removing question marker fragment (when) and updating the verb tense

and form using a rule-based approach that we have defined: Q=died, princess Diana
4. Adding the short answer: Q=died; princess Diana; August 31, 1997
5. Generating the set of possible answer structures S: S=died princess Diana August 31, 1997; . August 31, 1997 died princess Diana; . princess Diana died August 31, 1997; . . . .
6. Evaluating the different answer structures using a LM: Best Structure = princess Diana died August 31, 1997
7. Generating possible missing word for structure∗ with BERT: Princess Diana died [missing word] August 31, 1997 (missing word = on)

Answer: Princess Diana died on August 31, 1997.

2.4 Summary

At the start of the chapter, we discussed about the recent work done in the field of question answering and various techniques used. We explored the works done by various authors in this area. We then explored recent summarization and MT techniques used in natural answering wherein we discussed basic NMT model, Neural attention model for summarization. We then discussed about the recent summarization technique Pointer-Generator Network in which we discussed the Baseline Sequence-to-Sequence Attention model and the Pointer-Generator Network used in Abstractive text summarization. Lastly we discuss very recent work related to this task which uses transformers to generate natural answers in question answering domain.

Chapter 3

Dataset Description

In this chapter, we will talk about the details of the dataset we have used in this project. First we will discuss about the open domain dataset extracted from SQuAD and HarvestingQA, then some details of the LG-Soft data extracted from product manuals will be discussed.

3.1 Open Domain Dataset

We used the recently open sourced dataset made from standard machine comprehension datasets such as SQuAD (Rajpurkar et al., 2016) and HarvestingQA (Du and Cardie, 2018) to create auto-annotated data. This provide us with questions and factoid answers which we use as input to our system. For the ground-truth, we automatically extract full-length answers from the passages of these datasets by applying auto annotation technique which will be discussed below. We used 300,000 samples (question, factoid answer, full-length answer) from SQuAD and HarvestingQA. Additionally, we used manually annotated 15000 samples from SQuAD of which 2500 are used for development, 2500 for testing and rest 310000 were augmented with the auto-annotated data.

3.1.1 Auto Annotation Technique

Creating datasets for any new task is a challenge since modern systems based on neural architectures requires a large amount of data to train. To make the data creation task scalable, most of our training data is automatically generated from SQuAD and HarvestingQA. For each questionanswer pair, author of the paper (Pal et al., 2019) automatically extract the target full-length answers from corresponding passages. We iterate over the sentences in the context passage that contain the factoid answer and select the one that has the highest BLEU score with the question, given BLEUscore >= 35%. Given the question-answer pair (Q, A) and the passage P, the full-length answer T is the sentence, S, in the passage:

$$T = argmaxBLEU(Q,S) \tag{3.1}$$

$$iffA\epsilon S\&BLEU(Q,S) >= 35\%$$

This method of automatically extracting samples from existing QA datasets is scalable and can be reproduced with any modern QA datasets to generate more samples to augment this autogenerated samples extracted from HarvestingQA This autogenerated data samples follow a similar question distribution as SQuaD and is biased towards what" and "who" questions.

3.1.2 Manually Annotated Data

The auto-generated samples contain extra information in the ground-truth full-length sentences which are not aligned with the question or factoid answer. To refine our dataset to be more attuned to questions and also to capture the variability humans bring when generating new sentences, we manually annotated 15000 QA pairs, from the SQuAD dataset. We used multiple ways to answer the same question, such as in active and passive voice, to incorporate more variation to the target sentences. Apart from generating samples with the full-length answers well aligned with the question, we have also chosen complex samples from SQuAD which have long phrasal factoid answers to add more complexity to the data samples.

3.2 LG-Soft Dataset

This dataset contain 364 examples of domain specific data extracted from LG washing machine product manual guide through span based Q/A extraction technique. This dataset was manually annotated with the help of LG-Soft annotators to give the target full length answers. In general the questions were of the type of Frequently asked questions given in the manuals and some examples questions provided in between the manuals to explain things clearly to the user.

3.3 Dataset Creation

After going through the data we realized the available dataset is very noisy and contains grammatically incorrect questions as well as wrong or grammatically incorrect target answer in many cases.
In NLG systems, we know that there can be more than one correct answers but that is not incorporated well into the datasets. *eg*

Question *: Who is the ceo of google ?*
Factoid answer *: Sundar Pichai*
Target *: [Sundar Pichai is the ceo of google. , The ceo of google is Sundar Pichai.]*

In the existing dataset we see only A1 type annotations but A2 is also correct way to answer this question and should be added in the annotation. So we try to make the above improvements in the existing dataset by adding multiple correct alternate natural answers and remove some examples which doesn't make any sense. So we randomly selected around 7200 examples from 315000 train set, 420 examples NewsQA test set and made necessary addition in target answers and deleted examples not making any sense. Finally our dataset had 6768 and 380 examples. We also create a 166 examples of confirmatory (yes/no) QA containing different ways of asking questions. We will open source the improved data for further research.

3.4 Summary

In this chapter, we discussed some details about the dataset we have used in this project and techniques about how that dataset was created from existing span based Q/A dataset. Also auto annotation was discussed. The dataset was made using both auto-annotation technique and manual annotation. Then LG-Soft dataset details are given briefly. Lastly we discuss about the dataset we created by improving existing available datasets.

Chapter 4

Modified Pointer Generator Approach

In this chapter we will study about the modified Pointer Generator Approach used to solve the task of converting factoid answer to a full length answer proposed by (Pal et al., 2019) . We will discuss about the architecture details of the approach and discuss how exactly pointer generator is used to solve this task. We will also discuss some results of this approach and talk about error analysis of this model towards the end of this chapter

4.1 Architecture

The problem of generating full-length answer from the question and the factoid answer was framed into a Neural Machine Translation (NMT) task using two approaches. We built a model based on the pointer-generator architecture described in (See et al., 2017) except we use two encoders on the source side to encode question and factoid answer separately as shown in Figure 3. Let the question be represented by words $Q = q_1, q_2, ..., q_n$. Let the factoid answer be represented by words $A = a_1, a_2, a_3, ..., a_m$. The question and answer sequence are encoded using two 3-layered bidirectional LSTMs which share weights. This produces two sequences of hidden states

$$h_Q^t = BILSTM(h_Q^{t-1}, q_t) \tag{4.1}$$

$$h_A^t = BILSTM(h_A^{t-1}, a_t) \tag{4.2}$$

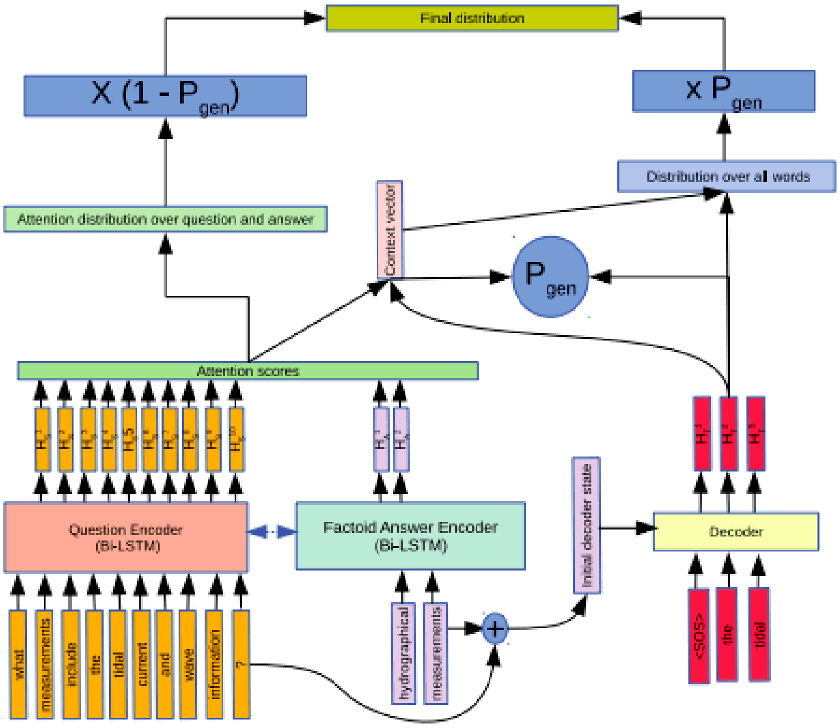


Figure 3: The 2 encoder pointer generator uses the question and factoid answer as input to generate a full-length answer in an end-to-end learning environment.

Figure 4.1: Modified Pointer Generator Architecture (Pal et al., 2019)

We choose to encode the source sequences separately, since there is no syntactic connection between the question and the factoid answer. We then stack together the encoded hidden states of the 2 encoders to produce a single list of source hidden states, $h_S = [h_Q; h_A]$. The decoder is initialized with the combined final states of the two encoders as

$$h_T^0 = h_A^m + h_Q^n \tag{4.3}$$

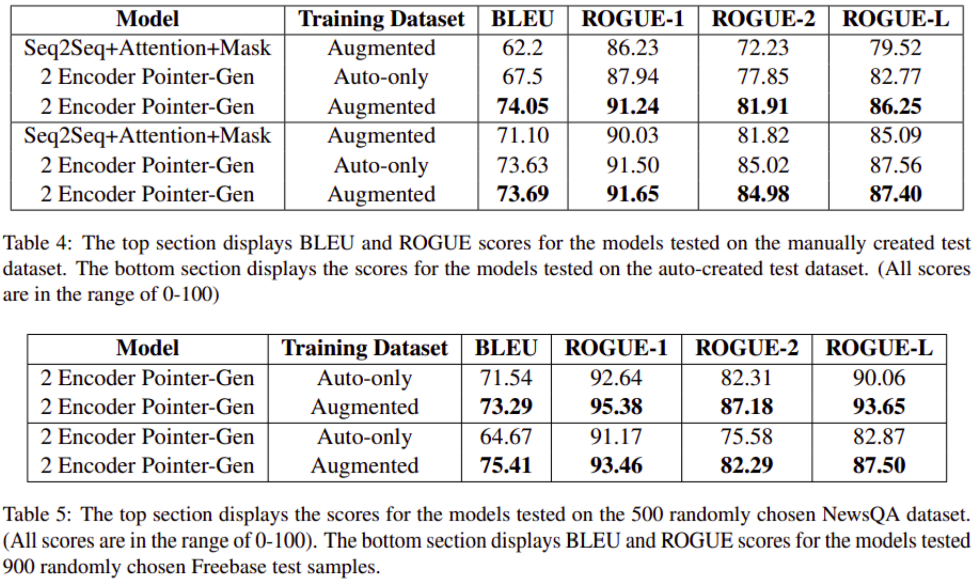
The challenge to correctly reproduce factual information in the full-length answer led us to use copy attention from the pointer generator network as described in (See et al., 2017). The copy distribution, using an extended vocabulary comprising of source words, will capture the probability of replicating words from either the question or answer, whereas the global attention distribution has the ability to generate new words from the vocabulary. The final probability of predicting a word is as follows:

$$P(W_{final}) = p_g P_{gen} + (1 - P_g) P_{copy} \tag{4.4}$$

Above is the final probability of generating a word . For out-of-vocabulary words which are present only in the source and for w belongs to V , only P_{copy} is used predict the word. These words are usually factual information from the question or answer, such as dates and named entities and hence needs to be copied exactly as it appears in the source sequences. Prepositions, conjunctions and other placeholders, such as $at, between, in$, which help in combining the question and answer sequences are usually in-vocab words not present in the source are predicted with P_{gen}. For in-vocabulary words which are present in the source, the final probability of predicting the word uses both the terms of above equation.

4.2 Results

In this section, we will illustrate the results of the Modified Pointer Generator Approach on the open domain dataset and other test datasets created from other question answering dataset like Freebase and NewsQA which is used to check cross dataset accuracy of the model.



Model	Training Dataset	BLEU	ROGUE-1	ROGUE-2	ROGUE-L
Seq2Seq+Attention+Mask	Augmented	62.2	86.23	72.23	79.52
2 Encoder Pointer-Gen	Auto-only	67.5	87.94	77.85	82.77
2 Encoder Pointer-Gen	Augmented	**74.05**	**91.24**	**81.91**	**86.25**
Seq2Seq+Attention+Mask	Augmented	71.10	90.03	81.82	85.09
2 Encoder Pointer-Gen	Auto-only	73.63	91.50	85.02	87.56
2 Encoder Pointer-Gen	Augmented	**73.69**	**91.65**	**84.98**	**87.40**

Table 4: The top section displays BLEU and ROGUE scores for the models tested on the manually created test dataset. The bottom section displays the scores for the models tested on the auto-created test dataset. (All scores are in the range of 0-100)

Model	Training Dataset	BLEU	ROGUE-1	ROGUE-2	ROGUE-L
2 Encoder Pointer-Gen	Auto-only	71.54	92.64	82.31	90.06
2 Encoder Pointer-Gen	Augmented	**73.29**	**95.38**	**87.18**	**93.65**
2 Encoder Pointer-Gen	Auto-only	64.67	91.17	75.58	82.87
2 Encoder Pointer-Gen	Augmented	**75.41**	**93.46**	**82.29**	**87.50**

Table 5: The top section displays the scores for the models tested on the 500 randomly chosen NewsQA dataset. (All scores are in the range of 0-100). The bottom section displays BLEU and ROGUE scores for the models tested 900 randomly chosen Freebase test samples.

Figure 4.2: Results of Modified Pointer Generator Approach (Pal et al., 2019)

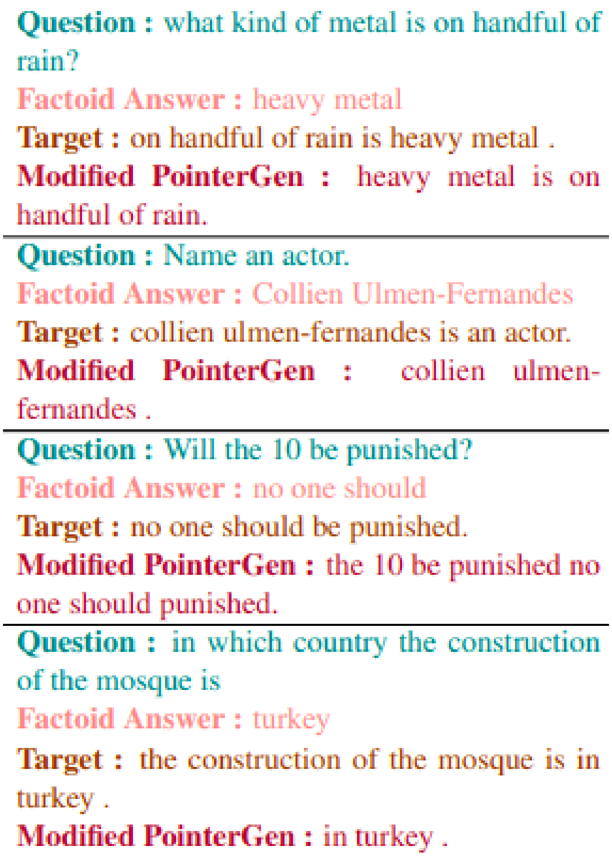
For that a test dataset is created having 900 examples from NewsQA and 500 examples from Freebase test samples.

Above figure contains 2 tables, the first table top section displays BLEU and ROGUE scores for the models tested on the manually created test dataset. The bottom section displays the scores for the models tested on the auto-created test dataset.

The second table top section displays the scores for the models tested on the 500 randomly chosen NewsQA dataset. The bottom section displays BLEU and ROGUE scores for the models tested 900 randomly chosen Freebase test samples. (All scores are in the range of 0-100)

4.3 Limitations and Error Analysis

The main limitation of this approach are stated in below points. Not all failure cases were of below type but for maximum cases these were the failure outputs. Also from the figure below we can see some examples wherein these failure cases are described in some detail. Also there were other failure cases



Question : what kind of metal is on handful of rain?

Factoid Answer : heavy metal

Target : on handful of rain is heavy metal .

Modified PointerGen : heavy metal is on handful of rain.

Question : Name an actor.

Factoid Answer : Collien Ulmen-Fernandes

Target : collien ulmen-fernandes is an actor.

Modified PointerGen : collien ulmen-fernandes .

Question : Will the 10 be punished?

Factoid Answer : no one should

Target : no one should be punished.

Modified PointerGen : the 10 be punished no one should punished.

Question : in which country the construction of the mosque is

Factoid Answer : turkey

Target : the construction of the mosque is in turkey .

Modified PointerGen : in turkey .

Figure 4.3: Failure cases of the Modified Pointer Generator Approach (Pal et al., 2019)

as well wherein the model just output the question itself which may be due to model becomes biased towards adding more part from the question than the factoid answer which results in complete copying of the question in some examples cases. Below are the main types of failure cases stated

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

In the above figure, Example 1 is from the Freebase dataset where the system confuses between the subject and the object. Example 2 is from Freebase not present in the training and validation data. Example 3 is from NewsQA dataset where the system fails to understand the semantics. Example 4 id from NewsQA dataset where the system fails to generate the complete full-length answer
In short this model doesn't give good results even for very straight forward example cases present in our dataset and so using it for general case queries would not be very beneficial as for many cases we will not get the expected target answer.

4.4 Summary

In this chapter we have discussed about the Modified Pointer Generator Approach in detail. We can discussed details about the architecture, results of the model implemented. Lastly we have also discussed about the limitation and error analysis of the model with the use of some examples.

Chapter 5

Finetuned DialoGPT-2 Approach

This chapter will describe about our second approach about the finetuning the DialoGPT model on the dataset discussed in previous chapter. We will disuss about how the DialoGPT model was finetuned how the input to the model was changed to solve the task of Natural Answer Generation. Then we will discuss about different experiments completed and discuss results of the experiment giving the best results. Also we will talk about some error analysis at the end of this section about this model and analyse why this models fails to give good results.

5.1 Brief introduction of DialoGPT model

DIALOGPT (dialogue generative pre-trained transformer) (Zhang et al., 2020) is a tunable gigawordscale neural network model for generation of conversational reponses, trained on Reddit data. Trained on 147M conversation-like exchanges extracted from Reddit comment chains over a period spanning from 2005 through 2017, DialoGPT extends the Hugging Face PyTorch transformer to attain a performance close to human both in terms of automatic and human evaluation in single-turn dialogue settings. The pre-trained model and training pipeline are publicly released to facilitate research into neural response generation and the development of more intelligent opendomain dialogue systems. DIALOGPT extends GPT-2 (Zhang et al., 2020) to address the challenges of conversational neural response generation. Neural response generation is a subcategory of text-generation that shares the objective of generating natural-looking text (distinct from any training instance) that is relevant to the prompt. Modelling conversations, however, presents distinct challenges in that human dialogue, which encapsulates the possibly competing goals of two participants, is intrinsically more diverse in the range of potential responses Like GPT-2, DIALOGPT is formulated as an autoregressive (AR) language model, and uses the multi-layer transformer as model architecture. Unlike GPT-2, however, DIALOGPT is trained on large-scale dialogue pairs/sessions extracted from Reddit discussion chains. Our assumption is that this should enable DIALOGPT to capture the joint distribution of P(Target, Source) in conversational flow with finer granularity. In practice, this is what we observe: sentences generated by DIALOGPT are diverse and contain information specific to the source prompt, analogous what GPT-2 generates for continuous text.

5.2 Architecture of DIALOGPT model

DIALOGPT model was trained on the basis of the GPT-2 (Budzianowski and Vulić, 2019), (Radford et al., 2018) architecture.The GPT-2 transformer model adopts the generic transformer language model (Vaswani et al., 2017) and leverages a stack of masked multi-head self attention layers to train on massive web-text data. The text generated either from scratch or based on a user-specific prompt is realistic-looking. The success of GPT-2 demonstrates that a transformer language model is able to characterize human language data distributions at a fine-grained level, presumably due to large large model capacity and superior efficiency. This model inherits from GPT-2 (Radford et al., 2018), a 12-to-48 layer transformer with layer normalization, a initialization scheme that accounts for model depth that we modified, and byte pair encodings for the tokenizer. We follow the OpenAI GPT-2 to model a multiturn dialogue session as a long text and frame the generation task as language modeling. First concatenate all dialog turns within a dialogue session into a long text ended by the end-of-text token. Then the conditional probability of response generation given dialogue is written as product of series of conditional probabilities.

5.3 Experiments

Handling models like GPT-2 and DIALOGPT which are trained on huge corpus of data is very difficult. Finetuning these models is even more difficult as these models require a very available memory to run on the machine. We were able to run all our experiments on Google Collab as on CFILT servers memory available was less. Hence in collab due to frequent session disconnection in between we managed to run few experiments that too decreasing the amount of data used for fine tuning. We tried different combinations of data for finetuning like combining manual annotated data with the auto annotated examples etc. But finally best results were achieved when only the manually annotated data was used to finetune the DIALOGPT model. Using the 13000 manually annotated and finetuning it for 8 epochs gave the best results on 420 examples of NewsQA dataset. Finetuning was done without changing the hyperparameters given in DialoGPT2 paper.

- **How the input was provided in finetuning the DGPT model?** Normally DIALOGPT models are used to make conversational chatbots and their finetuning is also done for making conversational agent where the input is the question asked and all the previous dialogues are kept as series of context and are passed as input to the model for training. Here in our task since all our questions are independent we have passed the input as question and the factoid answer as context and response as the GOLD standard answer that we have. Here instead of a series of conversations passed as context normally in our case we only give the factoid answer as a context to the model.

5.4 Results and model Output

In the above 2 tables we have evaluated finetuned DGPT model and compared the ROUGE and BLEU scores of this model with the modified pointer

Model	BLEU	ROUGE-1	ROUGE-L
DGPT finetuned on 13k manual data (8 epochs)	40.13	70.61	67.01
DGPT finetuned on 15k manual+auto (3 epochs)	33.77	59.27	53.68
Modified Pointer Generator	73.29	95.38	93.65

Table 5.1: DGPT Model results on 420 examples from NewsQA dataset

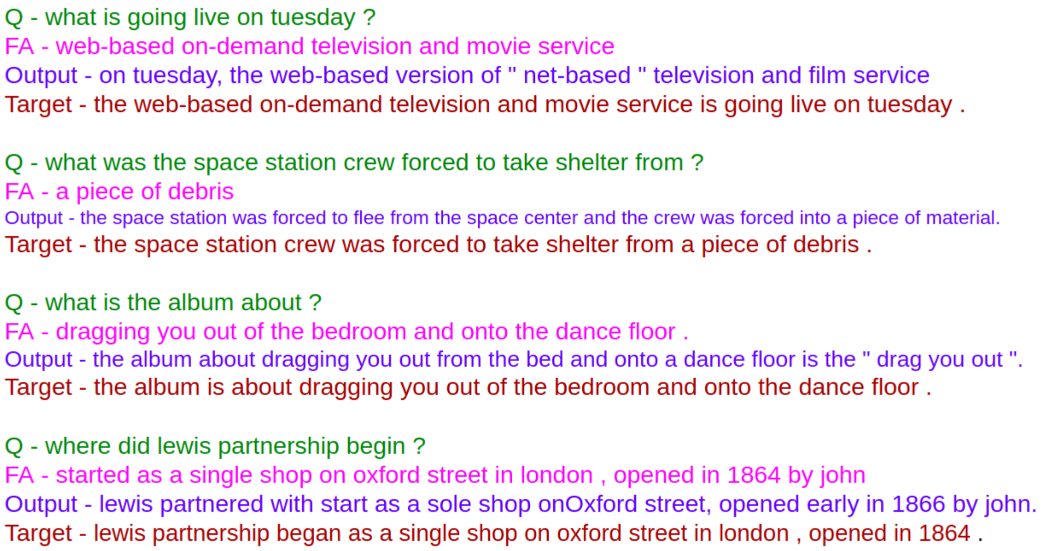
Model	BLEU	ROUGE-1	ROUGE-L
DGPT finetuned on 13k manual data (8 epochs)	33.23	66.51	60.35
Modified Pointer Generator	74.05	91.24	86.25

Table 5.2: DGPT Model results on 3200 examples from manually annotated test dataset

generator approach. Here we see that there has been a significant difference in all the 3 scores between the DGPT model and the Modified Pointer Generator approach. Scores on the output of DGPT model is very less. In the next section we will discuss some reasons why scores are very less in DGPT model.
Here we also discuss some of the output predictions on some examples by the DGPT model so that we get some idea of the model behaviour and how the answers are predicted.

5.5 Model Limitations and Error Analysis

Problem of adding unwanted things in the final answers which doesn't have any mention in the question and the factoid answer is the main shortcoming of this model. There are instances where there is repetition of some words in the answer and in some cases Factoid answer is not present in the final answer Mismatch in the questions having some numerical data or year is mentioned The model has some errors copying the proper nouns as given in the questions. The final answer has that names but with changed spelling. (eg:- elizabeth -> elizabetha; alexander -> alexandrick).
In the above figure also it is evident that many unwanted things are added in the output from the DGPT model as compared to the GOLD standard which



Q - what is going live on tuesday ?
FA - web-based on-demand television and movie service
Output - on tuesday, the web-based version of " net-based " television and film service
Target - the web-based on-demand television and movie service is going live on tuesday .

Q - what was the space station crew forced to take shelter from ?
FA - a piece of debris
Output - the space station was forced to flee from the space center and the crew was forced into a piece of material.
Target - the space station crew was forced to take shelter from a piece of debris .

Q - what is the album about ?
FA - dragging you out of the bedroom and onto the dance floor .
Output - the album about dragging you out from the bed and onto a dance floor is the " drag you out ".
Target - the album is about dragging you out of the bedroom and onto the dance floor .

Q - where did lewis partnership begin ?
FA - started as a single shop on oxford street in london , opened in 1864 by john
Output - lewis partnered with start as a sole shop onOxford street, opened early in 1866 by john.
Target - lewis partnership began as a single shop on oxford street in london , opened in 1864 .

Figure 5.1: DGPT model sample output examples

are not present in both the question as well as the factoid answer. In the first example we can see web-based changes to ”net-based” movie becomes film in the DGPT output. In the second example space station becomes space centre the complete answer structure is semantically wrong, the last part is making wrong sense. In the third example the term ”drag you out” is unwanted and is added at the end of the answer even when the factoid answer was copied in the output before, this shows that there are changes of repetition that occurs in the model output. In the fourth example we can see mismatch in the year mentioned in the factoid answer. In DGPT model output 1866 is given whereas actually the year given in the answer is 1864. So by analysing above examples we can conclude that DGPT model is not able to copy fact, numbers present in the question or the factoid answer. Also DGPT model has high bias of generating related things in the natural answer that is why BLEU and ROUGE scores decrease as GOLD standard doesn’t related words into account.

5.6 Summary

In this chapter we have discussed about the DIALOGPT approach and how it is finetuned to solve the task of natural answer generation. Then we discussed about evaluation scores of the DGPT model and sample output on some examples were discussed in details. We then discussed about the limitations and failure cases of the DGPT model and reasons of this failure were discussed towards the end of this chapter.

Chapter 6

Rule Based Approach

In this chapter, we will discuss the rule based approach to generate natural answers. We will first discuss about how this approach came to our mind, then we will discuss the algorithm. Then we will discuss results and model output using this approach and compare the results from previous approaches. Lastly we will discuss some error analysis of this approach.

6.1 Ideation

This approach came into our mind when we manually saw a large number of test examples and from that we were able to find a pattern in the full length answers. Then we came up with the idea of implementing the rule based approach which will use the sentence structure of the question at its core to generate a full length target answer. Initially we started with a very basic algorithm where we just replaced the WH words with the factoid answer and give the answer as it is without structuring but it had numerous failure examples. Then after seeing the failure examples of the above algorithm we were able to find a pattern related to the position of auxiliary verb and the main verb and then using this idea we were able to improve upon the failure cases we had in the above rule based approach. With this improvements many failure cases became exactly correct or very close to the target answer in the test dataset. So this is how this approach was formulated based on analysing the test examples and using the parse tree of the questions we were able to implement this approach.

6.2 Approach

There are 2 versions which will be discussed in this section wherein the second version is an improvement over the first version.
In the First version of our Rule based approach we just replaced the WH word present in the question with the factoid answer. In this method first we will find the position of the WH word present in the question then replace that word with given factoid answer to give a natural answer. Remember that we have not changed the question structure only there is a replacement of on word with the factoid answer. The WH word was found by using the POS tags output of the given question. AllenNLP constituency parser output was used to get POS tags of every word of the question. If the tag is ”WP” or ”WRB” then we replace that word with Factoid answer. Some examples are stated below for better understanding of the approach:-

Question : *What is the capital of India?*
Factoid answe*r : Delhi*
Rule Based Output v1 : *Delhi is the capital of India*
Target answer : *Delhi is the capital of India*

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

In the second version, we modify the above approach based on the position of AUX VERB and MAIN VERB present in the question. We formulate the algorithm as to solve the problem of ordering of natural answer i.e answer followed by question or question followed by answer. So, we look if the main verb and auxiliary verb are together then factoid answer is replaced with

WH part same as done in first version and if not then we have to add factoid answer in the end. In the latter case we start our answer from the word after the auxiliary verb, then after all the words before the main verb is copied we add the auxiliary word present in the question then we copy the part of from the question from the MAIN VERB to the end and then at the end we add the factoid answer. In other words we use dependency parse tree to get AUX and VERB tag and check if they are together and added this condition to the existing rule based model if they are together we follow first version. If AUX and VERB tag are not together then we add factoid answer at the end of the question. If question does not have verb in it then we add all words after auxiliary word in the answer, then add auxiliary verb and finally the factoid answer is added at the end. Some sample example output using second version is stated below:-

Question _: What is the capital of India?_
_Factoid answe__r : Delhi_
Rule Based Output v2 _: the capital of India is Delhi_
Target answer _: Delhi is the capital of India_

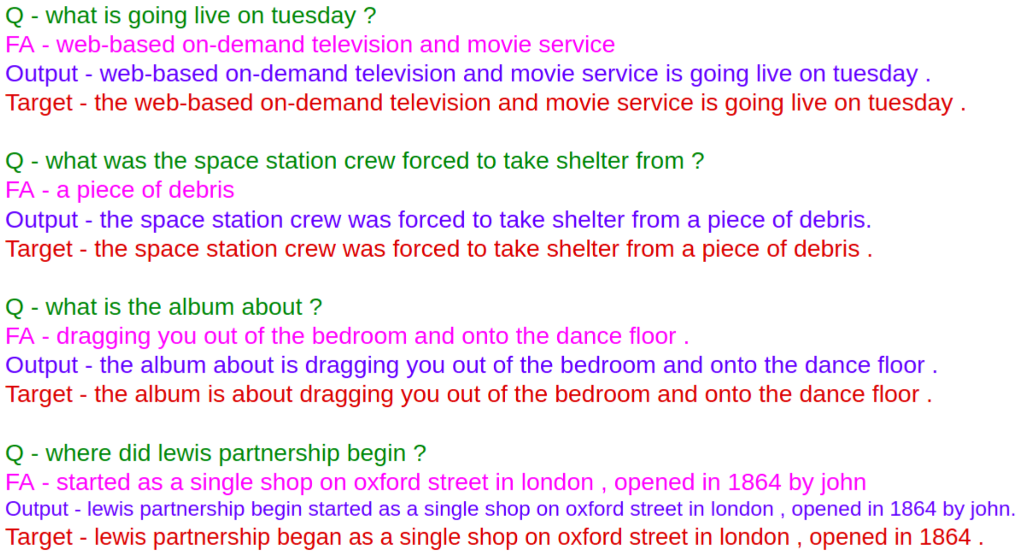
Question _: what was the space station crew forced to take shelter from?_
Factoid answer _: a piece of debris_
Rule Based Output v2 _: the space station crew was forced to take shelter from a piece of debris_
Target answer _: the space station crew was forced to take shelter from a piece of debris_

6.3 Results and Model Output

In the below table we have evaluated and compared the scores of all the models with rule based approach discussed above(R-1 means ROUGE-1 and R-L means ROUGE-L). We can see a significant rise in ROUGE-L score which means that higher order n-grams are matching the GOLD standard

Model	BLEU	R-1	R-L
DGPT finetuned 13k manual data (8 epochs)	40.13	70.61	67.01
Rule based approach v2	63.51	90.35	83.33
Rule based approach v1	69.59	89.166	72.177
Modified Pointer Generator	73.29	95.38	93.65

Table 6.1: Rule Based Model results on 420 examples from NewsQA dataset



Q - what is going live on tuesday ?
FA - web-based on-demand television and movie service
Output - web-based on-demand television and movie service is going live on tuesday .
Target - the web-based on-demand television and movie service is going live on tuesday .

Q - what was the space station crew forced to take shelter from ?
FA - a piece of debris
Output - the space station crew was forced to take shelter from a piece of debris.
Target - the space station crew was forced to take shelter from a piece of debris .

Q - what is the album about ?
FA - dragging you out of the bedroom and onto the dance floor .
Output - the album about is dragging you out of the bedroom and onto the dance floor .
Target - the album is about dragging you out of the bedroom and onto the dance floor .

Q - where did lewis partnership begin ?
FA - started as a single shop on oxford street in london , opened in 1864 by john
Output - lewis partnership begin started as a single shop on oxford street in london , opened in 1864 by john.
Target - lewis partnership began as a single shop on oxford street in london , opened in 1864 .

Figure 6.1: Rule Based v2 model sample output examples

but there has been a decrease of BLEU scores which points at lower n grams are not getting matched which may be due to restructuring of the question done hence there is a chance of some mistakes about handling all the TAGS in writing the algorithm. Also there are questions without the normal structure like question starting from auxiliary verb Eg. Can you help me in today's homework? etc. in the dataset which would decrease the scores.

6.4 Model Limitations and Error Analysis

This approach works reordering question sentence structure and copy pasting from the question and factoid answer and so if factoid answer is not factual based or is a clausal answer then this approach will fail. For eg last example of Figure 6.1 output answer had both began and started in it which is not right this is because the factoid answer contains a clause having verb part also in it, In our approach we are not checking the factoid answer structure to define our answers and hence for these examples this model will fail. Since the approach works on the question structure so if question is not properly well formed or incomplete then the answers will not be correct.

6.5 Summary

In this chapter we have discussed the rule based approach used to solve the task of natural answer generation. Though nowadays rule based approaches are very less used but here we have shown how a rule based approach is able to solve this task and works good for most questions. It is obvious that not all questions can be handled using rule based approach but we have shown the importance of rule based models in tasks like these and how they can be improved upon after analysing the output and finding some pattern. We have also discussed the error analysis of the model after the model results and some sample output examples.

Chapter 7

Post Processing Step : Grammar Correction Model

In this chapter we will describe about the post processing step in our rule based approach of natural answer generation. We will start with some introduction of the state of the art pre-trained transformer based Grammar Correction Model (GCM).

7.1 Introduction

Neural Machine Translation (NMT)-based approaches have become the preferred method for the task of GEC. In this formulation, errorful sentences correspond to the source language, and error-free sentences correspond to the target language. Recently, Transformer-based (Vaswani et al. (2017)) sequence-to-sequence (seq2seq) models have achieved state-of-the-art performance on standard GEC benchmarks. Now the focus of research has shifted more towards generating synthetic data for pretraining the Transformer-NMT-based GEC systems (Kantor et al. (2019), Grundkiewicz et al. (2019)). NMT-based GEC systems suffer from several issues which make them inconvenient for real world deployment: (i) slow inference speed, (ii) demand for large amounts of training data and (iii) interpretability and explainability; they require additional functionality to explain corrections, e.g., grammatical error type classification.

Hence to deal with the aforementioned issues by simplifying the task from sequence generation to sequence tagging. We used GECToR (Omelianchuk et al. (2020)) GEC sequence tagging system that consists of three training stages: pretraining on synthetic data, fine-tuning on an errorful parallel corpus, and finally, fine-tuning on a combination of errorful and error-free parallel corpora. This model gives state of the art results on the task of Grammar Error Correction on CoNLL-2014 and BEA-2019 datasets.

7.2 Experiments and Results

We have used 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. We have used Tesla T4 16GB GPU to carry out the experiments. For factoid questions, we use the 2 datasets having 380 and 6768 examples, results are given in table 7.1 and 7.2 respectively. For confirmatory questions we use 166 examples dataset created and formulate a rule based approach for confirmatory questions as well. As generally confirmatory questions has a structure AUX-NP-VP so using dependency analysis we formulate our answer as NP-AUX-VP. Results of confirmatory dataset is given in table 7.4.

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.*

As a post processing step of all our rule based approaches i.e. for factoid questions and confirmatory questions 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. So we carried our experiments using all 3 above encoder based GEC model as a post processing step in our rule based approach; In table 7.1, 7.2, 7.3, 7.4 : "RBV2+RoBERTa" means our rule based approach with grammar correction done by RoBERTa encoder and so on.

For DialoGPT using around 13000 manually annotated and fine-tuning it for 8 epochs gave the results on 380 examples of NewsQA dataset given in table 7.1.

Below example gives a qualitative comparison of output from different approaches explored in this paper. It is clear that our rule based approach (RBV2) with RoBERTa based GCM (RBV2+RoBERTa) achieves higher quality of natural answers as compared to MPG. Our developed approach gives comparable results in terms of BLEU and ROUGE-1,2,L scores with MPG and reduces inference time by 85%.

Model	BLEU	R-1	R-2	R-L	Avg. time(s)
MPG Pal et al. (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 7.1: Results on 380 examples of NewsQA dataset

Model	BLEU	R-1	R-2	R-L	Avg. time(s)
MPG Pal et al. (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 7.2: Results on 6768 examples of SqUAD dataset

Model	BLEU	ROUGE-1	ROUGE-2	ROUGE-L
MPG Pal et al. (2019)	64.1	85.7	72.5	78.8
RBV2	55.5	85.8	63.4	73.5
RBV2+BERT	54.8	81.9	60.4	71.4
RBV2+RoBERTa	55.6	82.6	61.3	72.0
RBV2+XLNET	54.8	82.0	61.1	71.7

Table 7.3: Results on 840 examples of Freebase 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 7.4: Results on 166 examples of Confirmatory questions dataset

7.3 Qualitative Analysis

Question *: where was the bus going ?*
Factoid answer *: phoenix , arizona*
MPG Pal et al. (2019) *: the bus going was at phoenix , arizona.*
RBV2 *[ours] : the bus was going phoenix , arizona.*
RBV2+RoBERTa *[ours] :The bus was going to Phoenix , Arizona.*
DialoGPT *[ours]: the bus was going to phoenix, anrizona.*

In the above example, MPG Pal et al. (2019) is making error in answer generation. Word position of was and going is interchanged and ”at” is added which is wrong, correct addition should be ”to”.
DialoGPT has changed arizona spelling to ”anrizona”.
RBV2 approach does give a answer but it is not complete word ”to” is missing from the answer which is added in the answer by a Grammar Error Correction (GEC) model GECToR with RoBERTa LM encoder. Omelianchuk et al. (2020).
This shows the importance of using GEC as a post processing step in our

rule based approach.

7.4 Error Analysis

This approach works reordering question sentence structure and copy pasting the factoid answer and so if factoid answer is not factual based or is a clausal answer then this approach will 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 type of grammar error by rule based approach is incorrect position of AUX word (*e.g.* is, are etc) in the answer which is not corrected by the Omelianchuk et al. (2020) in some cases. Also for questions starting with ”how many” word ”many” is added in the generated answer as well, which is wrong if the factoid answer extracted is a number. Here we rely on the GCM model to do the necessary corrections and the accuracy of GCM model in correcting this is good.

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

In the above example output answer had both begin and started in it which is not right this is because the factoid answer contains a clause having verb part also in it, In our approach we are not checking the factoid answer structure to define our answers and hence for these examples this model will fail. Since the approach works on the question structure so if question is not properly well formed or incomplete then the answers will not be correct.

7.5 Summary

In this chapter we studied the post processing step we used in our rule based approach. We talked about state of the art Grammar Correction Model used and have discussed results after using the above post processing step. Lastly we discuss qualitative analysis and error analysis of the proposed approach.

Chapter 8

Conclusion and Future Work

We saw the different approaches explored to solve the task of Natural answer generation, converting factoid answer to full length answer. In this chapter, we will summarize all the work described in the report. We will also highlight the important observations/conclusions from our work. At last, we will see the future work to be done on the project to accomplish the final objective of the project.

8.1 Summary

The objective of the project is to generate a natural answer given a question and its factoid answer.

The work done till now on the project is summarized below.

1. We started by describing the motivation behind the project. We explained why this task is important to solve and talked about some practical use cases. Then we discussed the problem statement of the project. Then we stated our approaches and the research plan followed during the project. Then we talked about our contribution by this project.

2. In the Literature survey, we explored recent ecent Summarization and Machine Translation techniques used in Neural Natural Answer Generation wherein we discussed basic NMT model and attention model

for summarization. Then we studied the Pointer Generator Network in which we covered the baseline model and the Pointer Generator model. Also we discussed a very recent work related to our problem statement in detail.

3. Next, we discussed about the dataset used in this project. Discussed 2 types of dataset. Also the technique used to generate this dataset from normal Question answering dataset was also discussed which can be used to augment the existing dataset with even more examples. Also we created and improved existing dataset.
4. Then, we discussed about the Modified Pointer Generator approach used to solve Natural Answer Generation task in detail. Results, Model Output, limitations and error analysis of the model was discussed
5. Then, we discussed about Finetuning DIALOGPT model approach used to solve Natural Answer Generation task in detail. Architecture, Experiments, Results, Model Output limitations and error analysis of the model was discussed
6. Then, we discussed about the Rule Based approach used to solve Natural Answer Generation task in detail. Ideation, Approach, Results, Model Output, limitations and error analysis of the model was discussed
7. Finally we discussed about the GEC model post processing step and talked how using this step improves the quality of the generated answers.

8.2 Conclusion

This section describes the key takeaways from our report.

- Our literature survey explored about the recent summarization and Machine Translation approaches useful to solve the problem task we have.

- We have explained the dataset details used with the technique used in creating the datasets. All the three approaches are explained giving the background and how they are implemented to solve the task of generating full-length natural answers given the question and the factoid answer.

- Covered neural as well as rule based approaches on a new dataset containing tuples of questions, factoid answers, and full-length answers of which 300,000 samples were automatically extracted and 15,000 samples were manually annotated.

- Using a pointer-generator Seq2Seq model leads to reasonable generation of full length natural answer but the sentence structure is not proper, DailoGPT model solves the fluency part but copy mechanism doesn't work well in this approach.

- Our newly proposed rule based approach with GEC model as a post processing step gives very comparable results with the state of the art approaches with 85% reduce in inference time. Answers generated from our proposed approach have better quality and very less chance of failing as compared to supervised approaches.

- GECToR with RoBERTa as encoder gives best results for our tasks. Also we notice there is a minor decrease in measured metrics after using GEC model, this is because the target answers manually annotated are not perfect and answers generated by our approach is also correct in maximum cases but since target answers are annotated not incorparating high quality grammar the scores are less.

8.3 Future Work

8.3.1 Deep Learning based models (Bilstm, transformers)

- Data Augmentation with indirect questions and ill formed questions, Equal distribution of all types of question and then retraining the model given in Pal et al. (2019)

- an end to end transformer based NLG model generating answers can be made by fine-tuning models solving abstractive summarization task like BART.

8.3.2 Rule Based Model

- Rules can be converted into features and a DL model can be fine-tuned on a pre-trained text generation model
- Augmenting rules which can handle some different kind of question like subordinate clause question, indirect questions etc

Evaluation metric BLEU, ROUGE was not appropriate for our task and not representative of model robustness on different type of questions. This is due to the fact that the generated answer has maximum parts from question and factoid answer. Other metrics like BERT-score, METEOR can be calculated. It will be nice if there can be evaluation metric specific to this task wherein exceeding length and missing factoid answer can be heavily penalised.

A hybrid model using both the modified pointer generator and our proposed rule based approach can be the best solution for the task of natural answer generation in the context of question answering.

Bibliography

Imen Akermi, Johannes Heinecke, and Frédéric Herledan. Tansformer based natural language generation for question-answering. In Brian Davis, Yvette Graham, John D. Kelleher, and Yaji Sripada, editors, *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, 2020. URL https://www.aclweb.org/anthology/2020.inlg-1.41/.

Abigail See, Peter J. Liu, and Christopher D. Manning. 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, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1099. URL https://www.aclweb.org/anthology/P17-1099.

Vaishali Pal, Manish Shrivastava, and Irshad Bhat. 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, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-5401. URL https://www.aclweb.org/anthology/D19-5401.

Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M. Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. Towards ai-complete question answering: A set of prerequisite toy tasks, 2015.

Jun Yin, Xin Jiang, Zhengdong Lu, Lifeng Shang, Hang Li, and Xiaoming Li. Neural generative question answering, 2016.

Shizhu He, Cao Liu, Kang Liu, and Jun Zhao. Generating natural answers by incorporating copying and retrieving mechanisms in sequence-to-sequence learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 199–208, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1019. URL https://www.aclweb.org/anthology/P17-1019.

Cao Liu, Shizhu He, Kang Liu, and Jun Zhao. Curriculum learning for natural answer generation. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4223–4229. International Joint Conferences on Artificial Intelligence Organization, 7 2018. doi: 10.24963/ijcai.2018/587. URL https://doi.org/10.24963/ijcai.2018/587.

Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. Pointing the unknown words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 140–149, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1014. URL https://www.aclweb.org/anthology/P16-1014.

Rongzhong Lian, Min Xie, Fan Wang, Jinhua Peng, and Hua Wu. Learning to select knowledge for response generation in dialog systems, 2019.

Yao Fu and Yansong Feng. 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, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1017. URL https://www.aclweb.org/anthology/N18-1017.

Siva Reddy, Danqi Chen, and Christopher D. Manning. Coqa: A conversational question answering challenge, 2019.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2016.

Alexander M. Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1044. URL https://www.aclweb.org/anthology/D15-1044.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 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, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1264. URL https://www.aclweb.org/anthology/D16-1264.

Xinya Du and Claire Cardie. 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, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1177. URL https://www.aclweb.org/anthology/P18-1177.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 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, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-demos.30. URL https://www.aclweb.org/anthology/2020.acl-demos.30.

Paweł Budzianowski and Ivan Vulić. Hello, it's gpt-2 – how can i help you? towards the use of pretrained language models for task-oriented dialogue systems, 2019.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2018. URL https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

Yoav Kantor, Yoav Katz, Leshem Choshen, Edo Cohen-Karlik, Naftali Liberman, Assaf Toledo, Amir Menczel, and Noam Slonim. Learning to combine grammatical error corrections. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 139–148, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4414. URL https://www.aclweb.org/anthology/W19-4414.

Roman Grundkiewicz, Marcin Junczys-Dowmunt, and Kenneth Heafield. Neural grammatical error correction systems with unsupervised pretraining on synthetic data. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 252–263, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4427. URL https://www.aclweb.org/anthology/W19-4427.

Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanskyi. 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, July 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.bea-1.16.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 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, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://www.aclweb.org/anthology/P02-1040.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W04-1013.