Text Generation

A Master of Technology Seminar Report

Submitted in partial fulfillment of requirements for the degree of

Master of Technology

By

Tathagata Dey
Roll No.: 22M0765

under the guidance of

Prof. Pushpak Bhattacharyya

Department of Computer Science and Engineering
Indian Institute of Technology, Bombay

APRIL, 2023

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 cause for disciplinary action by the Institute and can 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: 28/04/23 Tathagata Dey

Place: IIT Bombay, Mumbai Roll No: 22M0765

Acknowledgment

I would like to thank my project guide, **Prof. Pushpak Bhattacharyya** for giving me a valuable opportunity to work under his guidance. All the research update meetings have been extremely helpful in resolving doubts related to the project. I find myself lucky to have a chance to get the benefit of his immense knowledge, support and guidance. I am very grateful to my super-senior **Vishal Paramanik** and senior **Divyank Tiwari** from IIT BOMBAY for all the support and guidance provided through out the semester. I would also like to show my gratitude to all the members of **CFILT (Center for Indian Language Technology) lab, IIT Bombay**, for sharing their experience and wisdom and being an integral part of my project here.

Abstract

In this era of ChatGPT text generation is one of the booming and important prospects of machine learning and artificial intelligence. In the twenty-first century, text generation can be helpful to assist us from talking to chatbots to writing long and boring essays. Every piece of generated text deals with some useful techniques and also falls under the umbrella of Natural Language Generation.

Text generation now can entertain us from various different angles, thanks to the advanced digital lifestyle. Our work-related emails, letters, resumes everything can be handled with such a system. Even writing documentation, and instructions can be boring enough to get replaced with these high efficient systems. On the other hand, we often converse with chatbots on various platforms starting from e-commerce to even technical platforms.

Hence, studying and working on text generation can revolutionise our lives. This study, therefore, focuses on text generation as a part of natural language generation. The study goes through the background and different prospects of natural language generation and talks about a few cutting-edge research applications.

Contents

List of Figures

List of Tables

Chapter 1

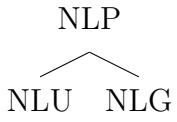
Introduction

1.1 Objective of Study

The objective of this study includes knowing various aspects of Natural Language Generation, why it is important and its systems. Then we go on to understand how an NLG system is built and what are the necessary conditions and objectives to take care of during this process. Furthermore, we dive into various tasks or use cases where NLG can be useful.
In the next phase, we learn about Script generation which is one of the most crucial cutting-edge research of the modern NLP world. Works of CFILT lab in this field and works from other researchers. Then we move on to Data-to-text generation and study its prospects and CFILT lab works.

1.2 Natural Language Generation

Natural Language Generation is a domain of Machine learning which is situated at the meeting point of Computer science, Linguistics and Probability. NLP contains NLU, i.e. Natural Language Understanding and NLG, i.e., Natural Language Generation.



Natural Language Generation is the field of Natural Language Processing which deals with the generation of new texts. Studying or processing texts is different from generating them. Because generation must include creativity, coherence, consistency and other factors. Hence, the objective of generation should be dealt with with special importance.
Generation can be for many different use cases which we will discuss in Chapter 2. However, we need to understand why NLG has to be computed differently from regular objectives. Typical NLG problem statements can be like producing a summary from a paragraph which we all did in our schools as summary writing. Also, in generating dialogues, i.e. given a sentence from a conversation, the objective is to generate a response. These tasks are different from regular NLP tasks, for example, sentiment analysis. In sentiment analysis, given a text, we have to determine what is the sentiment of the text, i.e. positive, negative or may be neutral. But, in generation tasks, not only understanding the input is sufficient, we have to train the model to identify what could be the response.
The modern NLP era is revolving around NLG only. Most of the cutting-edge research is from this domain. A very obvious example we can see in front of us is ChatGPT. There are many other objectives of NLG which we will also discuss later in this study.

1.3 Motivation

Natural Language Generation is one of the most impactful domains of Machine learning and Artificial Intelligence in the twenty-first century. The advent of ChatGPT has already shown the world what NLP systems can accomplish and how useful they can be in human assistance. Hence, learning about NLG can help us develop efficient and important systems for society. NLG can apply to many other prospects also. Some of them are elaborated on in the upcoming chapters.
Movie script generation is one such prospect which is a task which is time-consuming, dull and also can be automatable. It brings in a high economic

importance and resource constraints. Also, data-to-text generation is talked about in this study. In this data-driven world, spoon-feeding a complex piece of information to humans can give a lot of importance to such a system and bring high potential interest. So, being a cutting-edge research field, this should be studied with significant importance.

1.4 Road Map

In Chapter 2 background of Natural Language Generation is described. We see various objectives of NLG, building NLG systems and so on. Further in Chapter 3, we study Script Generation, what is the problem statement and what are the research works in this field. Similarly, in Chapter 4, we talk about Data-to-text generation and in Chapter 5, we conclude the study.

Chapter 2

Background of Natural Language Generation

2.1 Applications

Computer systems often generate or represent data which is vastly different from the human-comprehensible structure. Such as schedule databases of airlines or railway trains and busses, spreadsheet details of different accounts, query-key-value-based representation of medical reports and so on. These different forms of data are often important and necessary for a reader to comprehend quickly and understand the necessary information. However, most modern-day systems lack in this regard. A Natural language generation system comes into the picture here. NLG can help in generating a human-understandable piece of text which contains rich information and can be understood by a reader quickly.

Also, natural language generation can generate a whole story out of a given set of keywords or phrases, which is relevant in today's world. There are numerous different genres which can make good use of creative, coherent stories for different purposes. Summarizing a larger text into a small concise form can be helpful for readers in some cases too. Also, dialogues are of prime importance now as conversational AI is booming its territories into

real-world applications[1].

Natural language generation ideas can be useful to represent medical information in a more readable way [Buchanan et al., 1995][Cawsey et al., 1995], answer specific questions from a previously given object-base [Reiter et al., 1995], describe weather forecast reports [Goldberg et al., 1994] or even some form of statistical data into easily comprehensible way [Iordanskaja et al., 1992]. NLG can also help in summarisation [McKeown, 1985], writing job descriptions for firms [Caldwell and Korelsky, 1994], or even documentation for different programs [Paris et al., 1995].

2.2 Building a Natural Language Generation System

Building a Natural language generation takes several different parts to be incorporated into a single pipelined system. It usually starts with building a corpus to develop a system on [2]. The initial corpus should contain human-generated text for the model to better understand those patterns. The corpus must be annotated to determine the target output and input. Thus, it can be passed through all the different steps of the NLG system.

The system can be divided into multiple sub-tasks, however, this differentiation is not always clear depending on the objective of the systems. According to [Reiter and Dale, 1997] we can arguably divide the system into these six sub-tasks.

2.2.1 Content Determination

Content Determination deals with planning what information should be conveyed next. Typically the input information can be structured as *entity*, *relation*, *entity* format. These structured data are transferred into textual

[1] Conversational AI is a particular task-oriented model that can simulate human dialogues.

[2] A corpus is a collection text for a specific task on which the machine learning model has to be trained.

sentences by the system. *Reiter et al.* mentioned an example of representing information as data objects.

```
relation: IDENTITY
args[argument1: NEXT-TRAIN
     argument2: RAJDHANI EXPRESS
]
Generated Sentence: The next train is Rajdhani Express.
```

There can be other different ways to represent a content structure, however, converting them into text sentences needs sound determination of the information to be represented.

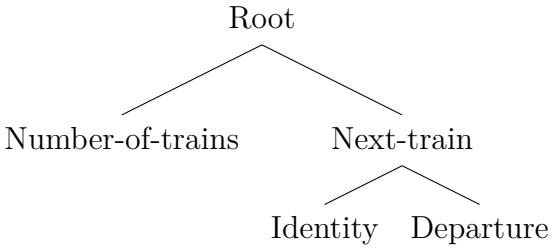
2.2.2 Discourse Planning

Discourse Planning deals with the structural ordering of the information to be conveyed in the sentence. Through content determination we can identify the information, however, any random collection of those information can not be represented as a sentence. The order of information can vary the meaning largely.

```
relation: Departure
args[departing-entity: Rajdhani Express
     location: Mumbai
     departure-time: 1800
]

relation: Number-of-trains
args[source: Mumbai
     destination: Delhi
     number: 5
]
```

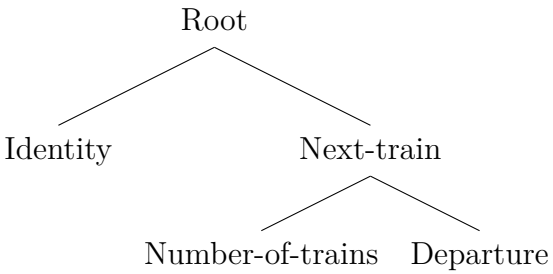
The above example is influenced by the paper [Reiter and Dale, 1997] where the author mentioned the importance of discourse structure. The three relations shown above can be structured in different ways. However, the right way to order them is as follows.



The generated sentence can be written as

> **There are 5 trains between Mumbai and Delhi, of which the next train is Rajdhani Express, leaving at 1800.**

However, we can see some other combinations such as,



wouldn't have been that good to express the information appropriately.

2.2.3 Sentence Aggregation

Sentence Aggregation is the next stage of building a system. When separate contents are decided and the ordering has been done through discourse planning, it passes through the aggregation stage. For example, we can consider the previous information. There can be two separate messages such as *Identity* and *Departure* which can be expressed together.

Through discourse planning, once we have decided the order, we may aggregate the two sentences and one sentence has to be generated. **The next**

train is Rajdhani Express, which leaves at 1800. is the aggregated sentence here.

Although aggregation may not be necessary in all cases, however many of often the semantic and pragmatic meaning is well expressed in a combined sentence than in some disjoint informative sentences. This can be argued over different languages and their unique patterns, but since we are taking *English* as the language of use here, it is well suggested.

2.2.4 Lexicalization

Lexicalization is the task of choosing the right word at a specific instance to express the meanings correctly. A particular meaning can be represented with different words, for example, we can consider a sentence *The train will reach the destination soon* or *The train will arrive at the destination soon*. Both have similar meanings however the words *reach* and *arrive* are different.

Lexicalization emphasizes the choice of words which is of great importance in generating a coherent piece of text.

2.2.5 Referring Expression Generation

Referring to expression generation deals with choosing the right word to refer to a definite entity. For example, the train *Rajdhani Express* can be referred to as *the Rajdhani Express*, which points to the same train.

Referring to expression generation may look similar to lexicalization, however, there is a subtle difference. Lexicalization primarily deals with choosing words to express a sentiment. On the other hand, referring to expression generation only deals with words to refer to an entity.

2.2.6 Linguistic Realisation

The linguistic realisation is similar to the surface realisation of text generation. The initial parts such as content determination and discourse planning decide the information to be conveyed and structure it. Then sentence aggregation helps in putting the information together into a sentential form.

Lexicalization and Surface Realisation determine the pronouns and the right words for the meaning.

At last, linguistic realisation adds the rules of language grammar to make a complete sentence. We can consider the example *There are 5 trains between Mumbai and Delhi, of which the next train is Rajdhani Express, leaving at 1800.* Here the words *of, at, is* are added by linguistic realisation. These words can be prepositions of some verbs which are denoted by the rules of grammar.

2.3 Different Tasks in NLG

Natural Language Generation is a growing field with vast different applications to real-world scenarios. NLG encompasses many problem statements which generate text as a bottom line. We are indeed going to talk about them. However, there are a few things to note before jumping right into the domains.

Although these separate tasks exist, there is no clear boundary to distinguish one from another. A problem statement may encompass different objectives of NLG simultaneously also.

The tasks can be classified as given in figure 2.1. The concepts of this classification have been taken from the paper [Dong et al., 2022]. However, there are correlations between these tasks. Such as summarization can be incorporated into distractor generation or text expansion can be used in dialogue generation and so on. Hence, we cannot clearly distinguish them from each other.

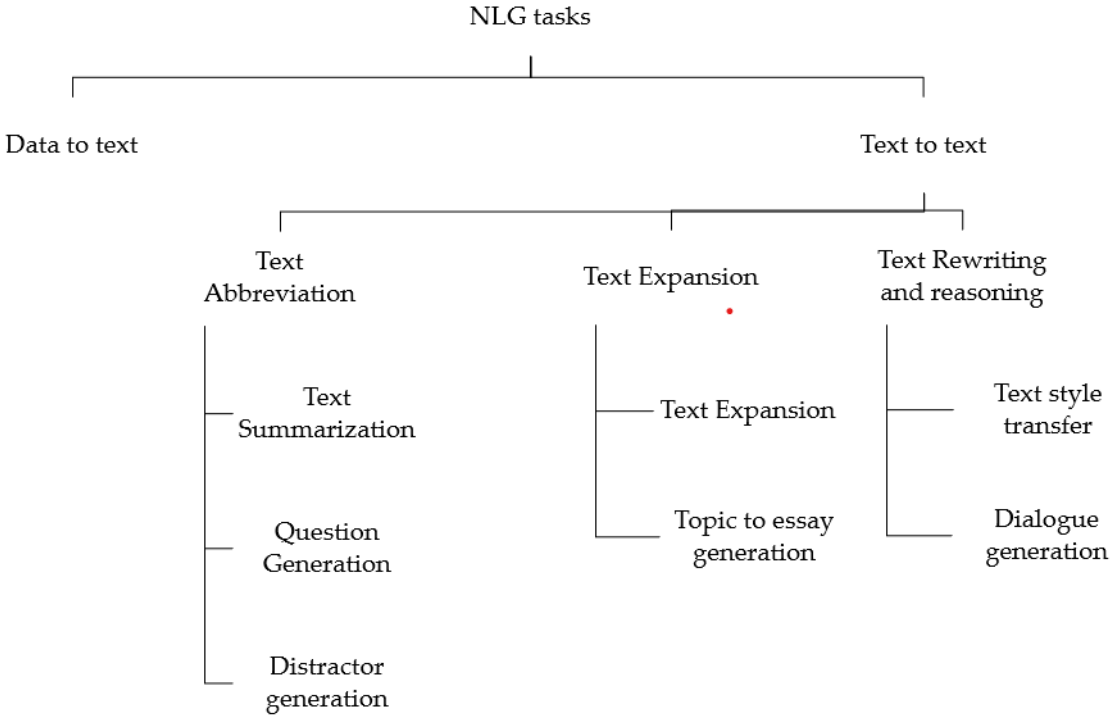


Figure 2.1: Different Tasks under Natural Language Generation

2.3.1 Data-to-text Generation

Data-to-text Generation is one of the most profound problems of natural language generation. Data is found in abundance in today's world. Starting from summary or cars passing through a section of road to cosmic microwaves found in the universe, everything is measured and noted as a form of data. Data is also highly efficient to measure scientific theories or even finding patterns in natural event occurrences. Since the advent of the data-driven world, the efficiency of scientific systems has increased to quite a few extents.

However, till now we haven't seen a correlation between having a data-driven world with natural language generation. Well, as it seems, with access to an increased volume of data we also intend to understand them. Data shouldn't only be understood and used in productivity by professionals. So,

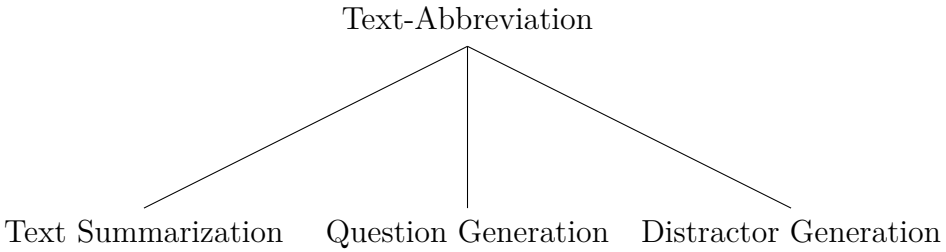
there comes a need to represent data in an easily understandable form or natural human language.

We may have data in the format of table, or in the form of RDFs [3] or even in the form of graphs or images. The objective is to convert them into natural language.

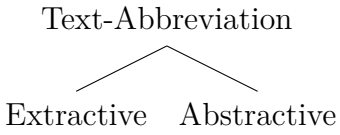
The usefulness of Data-to-text generation can be seen in many modern-day scenarios. Understanding weather forecast graphs, medical reports, any form of engineered statistical data, medical images, star chart observation data and many more can be made simpler with natural language generation.

2.3.2 Text Abbreviation

The primary goal of text abbreviation is to filter out key information from a text and represent it in a precise form. There can be many different objectives of this task, based on which the field is classified into three different tasks.



On the other hand, based on the method of text abbreviation, it can be classified into two separate ways.



Text Summarization

To understand the difference between extractive and abstractive, let us consider the example of text summarization. Summarization is a simple enough

[3] RDF is Resource Description Framework. Elaborated in Chapter 5.

task as the name suggests. Given a piece of text and the degree of summarization, the objective is to generate a concise text which depicts the meaning and information of the given text.

Here, extractive text summarization refers to the idea of picking out sentences from the input text which are rich in information and inevitable to express the meaning. The number of sentences picked depends heavily on the degree of summarization or compression ratio.

On the other hand, Abstractive text summarization deals with understanding the complete information and meaning of the text and expressing it with new sentences and sets of words in a more concise way. Abstractive text summarization requires high computation. With the advancement in computational power and deep learning frameworks, abstractive summarization nowadays generates a finer output of the task.

There are many datasets available to perform a text summarization task. *CNN/DailyMail* [Hermann et al., 2015], *New York Times articles* [Paulus et al., 2017], *Gigaword* [Rush et al., 2015] are some of the popular ones. *BART* [Lewis et al., 2020], *MASS* [Song et al., 2019], *PEGASUS* [Zhang et al., 2020] models are extensively used in summarization.

Question Generation

Question generation deals with forming questions from a given paragraph. The questions can be information based or subjective, depending on the training objective. This task has a humongous use case in today's growing educational prospects. *SQuAD* [Rajpurkar et al., 2016] and *MS MARCO* [Bajaj et al., 2018] datasets are available to build a question generator model.

Distractor Generation

Distractors are useful in generating similar options to an answer or relative piece of information. *RACE*[Lai et al., 2017] is a popular dataset for distractor generation.

2.3.3 Text Expansion

Text expansion is pretty much the opposite task of text summarization. In text extraction, the objective is to elongate a text. Often text expansions are useful to generate paragraphs that are easier to read, understand and comprehend. Information-rich content can be elaborated with expansion. Also, given a summary, we can generate an essay with text expansion.

Short text expansions deals with generating a paragraph from a short paragraph. Mostly, it elaborates on the mentioned information. There are datasets available such as *Wikipedia* [Tang et al., 2017] and *Fiction Corpus* [Safovich and Azaria, 2020] to develop such a model.

Topic-to-essay Generation

Topic-to-essay generation is a task to generate essays from a given topic. *Stanford Sentiment Treenbank dataset* [Socher et al., 2013] contains movie reviews which is used for this task. There are also *Customer Reviews* [Hu and Liu, 2004] and *Beer Reviews* [McAuley and Leskovec, 2013] datasets available. An example of topic to essay generation is shown in figure 2.2 from the paper [Feng et al., 2018].

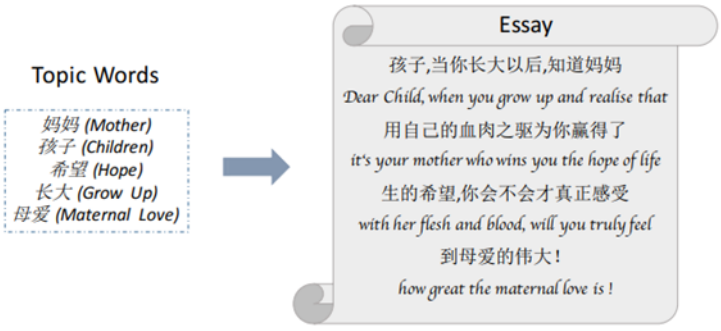


Figure 2.2: Example of Topic to essay generation

2.3.4 Text Rewriting and Reasoning

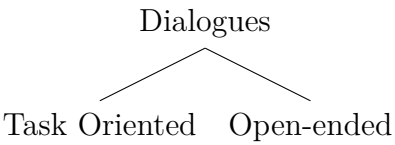
Text-to-style Transfer

Style transfer deals with changing the style of the text without disturbing the primary content and information of the text. Paraphrasing uses style transfer heavily. *Yelp Reviews* [4] and *Amazon Food Reviews* [McAuley and Leskovec, 2013] are two datasets developed with customer reviews. These datasets are used for this purpose.

Dialogue Generation

Dialogue generation is one of the most popular and crucial tasks in the modern world. From the very beginning of NLP and AI progress, chatbots have started to come into our daily picture. Starting from customer assistant chatbots on e-commerce platforms to ending with today's chatgpt, dialogue generation has always been a profound problem.

A sound dialogue generator system creates engaging, relevant conversational sentences which are also coherent and consistent. The conversation is intended to follow the Gricean Maxim rules [Okanda et al., 2015]. Dialogues are of two types as shown below.



Task-Oriented dialogues converse mostly to accomplish a goal. For example, customer assistant bots talk about a particular objective, i.e. the issue that the customer is facing. However, talking to the chatgpt of Google assistant feels different. Even when there is no definite objective of the conversation it may continue. These conversations are regarded as Open-ended dialogues.

There exists *Daily Dialogue* [Li et al., 2017] dataset with multi-turn dialogues [5]. Also *Persona Chat* [Zhang et al., 2018] dataset can be used for the

[4] http://www.yelp.com.dataset/

[5] multi-turn dialogue is a conversation which contains multiple speakers.

same. This dataset is a collection of personal small-talk conversations.

2.4 Tools Used in Building NLG Systems

Accomplishing a natural language generation task requires specific tools and frameworks. Starting from recurrent neural networks, the world of deep learning has progressed a lot to develop some high computational architectures. These architectures have also been proven to be much more efficient than previous techniques. Some of those ideas are discussed below.

2.4.1 Encoder-Decoder Architecture

Encoder-Decoder architecture is considered to be a significant improvement over Recurrent Neural Networks as it can remember long contexts.

Encoder comprises a unit (LSTM or GRU in many cases) which takes sequential inputs and goes through the whole text. The encoder is designed to understand the meaning or read the whole input text and generate a matrix containing the whole information. This matrix is then passed on to the decoder.

The decoder then generates predicted words with the matrix input and previous token as output. A visual representation of the encoder-decoder framework is shown in 2.3.

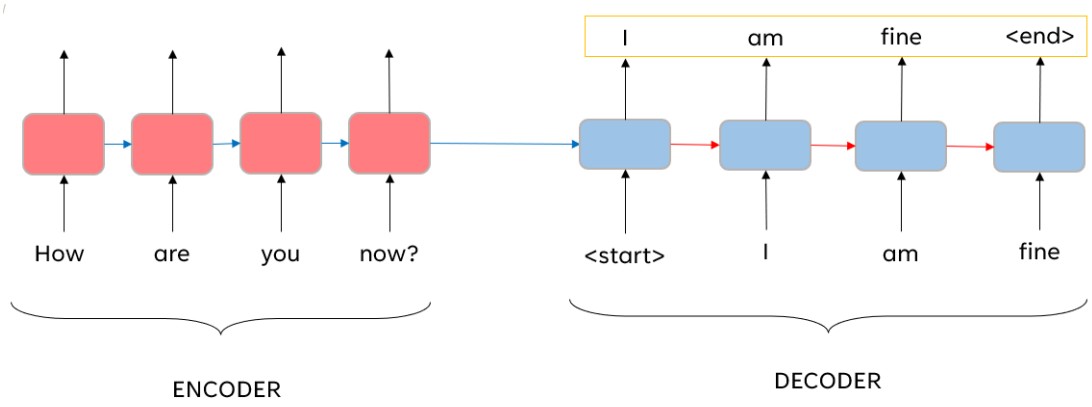


Figure 2.3: The Encoder-Decoder Architecture

2.4.2 Transformer

The transformer is the protagonist of modern natural language processing. From the revolutionary paper [Vaswani et al., 2017], the world of NLP found the new age artificial intelligence. Transformers are based on an encoder-decoder architecture with an attention mechanism.

In Transformer the encoder and decoders are designed with a multi-head attention mechanism and feed-forward networks. The architecture is shown in 2.4 which is taken from the paper [Vaswani et al., 2017].

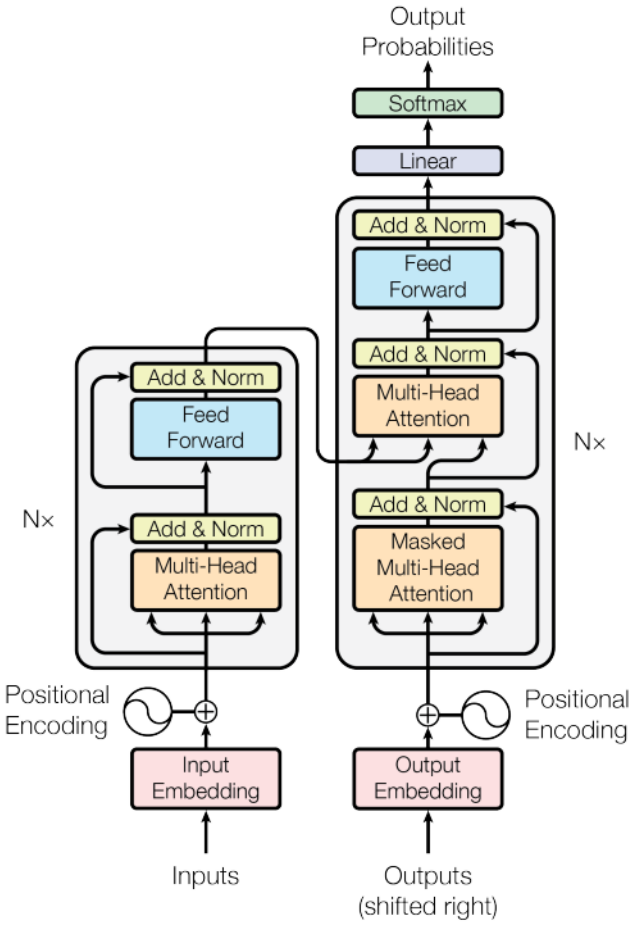
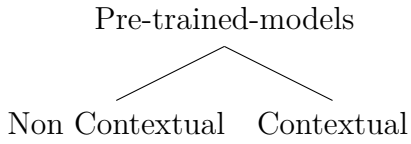


Figure 2.4: The Transformer Architecture

2.4.3 Pre-trained Models

Pre-training is a strategy to train a model on a given corpus beforehand. The corpus chosen is usually a general purpose corpus which helps the model to understand and learn basic rules and tactics of the language and also get a brief idea about the specific task. There are many popular language models which are pre-trained such as BART [Lewis et al., 2020], GPT [Brown et al., 2020] and so on.

Now pre-trained models can be of two types.



Non-contextual pre-training can happen when the training data isn't context-dependent. Such as Word2vec [Mikolov et al., 2013] is trained to define the word vector of any word. However, it doesn't intend to find the important or meaning of that word in a sentence.

On the other hand language models are contextual as they primarily focus on Natural Language Understanding (NLU). These models understand the given input text and can modify the output based on separate task objectives (which is called fine-tuning).

2.4.4 Evaluation Metrics of NLG Systems

Like any other Artificial Intelligence field, NLG also requires precise evaluation of the generated texts. However, new text generation may not always be completely evaluated through some statistical parameters and numbers. Hence, the importance of Qualitative evaluation or human evaluation is necessary and can be seen in most of the state-of-the-art works.

Some of the popular metrics used are Bilingual Evaluation Understudy or BLEU [Papineni et al., 2002]. BLEU calculates the co-occurrence frequencies of n-grams between two sentences.

Another popular metric is Recall-Oriented Understudy for Gisting Evaluation or ROUGE [Lin, 2004]. ROUGE emphasizes the recall score between generated and reference text. There are different kinds of this metric such as, ROUGE-n measures the co-occurrence of n-gram, ROUGE-l measures the longest common subsequence and so on.

On the other hand, the human evaluation focuses on grammatical correctness and creativity. The coherence[6], faithfulness[7], consistency[8] and lexicalization[9] can only be judged by a human.

[6]Coherence identifies if the generated text is logically valid
[7]Faithfulness judges if the generated text follows the input text
[8]Consistency refers to being faithful to previously generated text
[9]Lexicalization deals with identifying if right words have been chosen at right places

Chapter 3

Script Generation

3.1 Problem Statement

Script Generation as a whole can incorporate various use cases. Such as scripts for movies, plays, dramas and much more. However, due to the larger audience, impact and resource involvement movie script generation can be a useful problem statement from many perspectives.

Writing a movie script typically starts from generating a key idea, on which the plot has to be developed. This idea may be represented with a set of keywords or even a couple of sentences. The genre of the movie may also be a feature of this set. Therefore a plot can be generated based on this set of keywords and later the plot can be converted into scenes.

The example of 3.1 is taken from a dataset, collected by the CFILT Lab, IIT Bombay. This shows a scene from the movie ”12 Monkeys” and the specific format and rules to depict a scene soundly. The goal of this task is to generate scripts for a movie, given a set of keywords.

FADE IN:

INT. CONCOURSE/AIRPORT TERMINAL - BAY

CLOSE ON A FACE. A nine year old boy, YOUNG COLE, his eyes wide with wonder. watching something intently. We HEAR the sounds of the P.A. SYSTEM droning Flight Information mingled with the sounds of urgent SHOUTS, running FEET, EXCLAMATIONS.

YOUNG COLE'S POV: twenty yards away, a BLONDE MAN is sprawled on the floor, blood oozing from his gaudy Hawaiian shirt.

A BRUNETTE in a tight dress, her face obscured from YOUNG COLE'S view, rushes to the injured man, kneels beside him, ministering to his wound.

ANGLE ON YOUNG COLE, flanked by his PARENTS, their faces out of view, as they steer him away.

FATHER'S VOICE (o.s.)
Come on, Son --this is no place for us.

YOUNG COLE resists momentarily, mesmerized by the drama.

YOUNG COLE'S POV: intermittently visible through a confusion of FIGURES rushing through the foreground, the BLONDE MAN reaching up and touching the cheek of the kneeling BRUNETTE in a gesture of enormous tenderness, a gesture of farewell, while the P.A. SYSTEM continues its monotonous monotone...

Figure 3.1: Example of a Movie Script from the movie ”12 Monkeys”.

The problem statement is divided into two subtasks. The first one is **Plot Generation** and the second one is **Scene Generation**. The objective of the plot generation task is to produce the plot or story of the movie given the set of keywords. Then the plot goes through the second stage, i.e., the scene generator. Here, the plot is converted into well annotated and structured script for the movie. An end-to-end model to perform such extensive and high computational work is ambitious to achieve. Hence, this staged generation works much better. The architecture is better shown in figure 3.2.

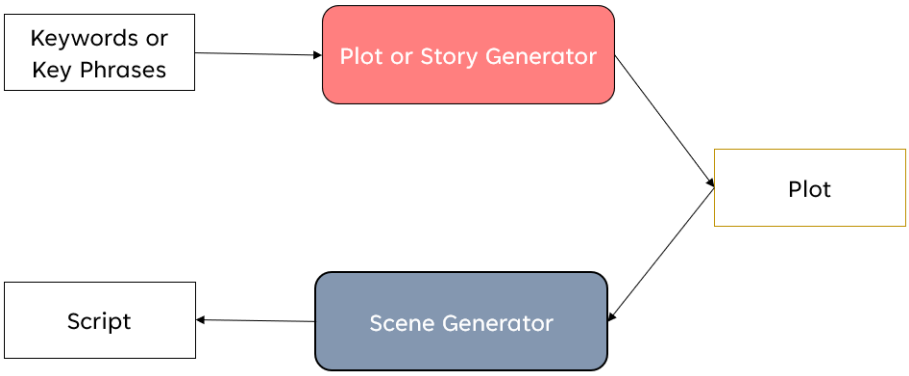


Figure 3.2: Stages of Movie Script Generation

3.2 Motivation

Movies are one of the greatest sources of entertainment all around the world. The movie industry is also of high economic valuation with some movies earning over a billion USD. Good movies also leave a long-lasting impact on society. A movie script typically is 30,000 words long which can be related to a 100 pages book. Much of the long script is taken by headings, character names, transitions and other annotations. So, writing these scripts can be dull, monotonous, energy and time-consuming.

With the advent of transformers and high-computing machines, story generation through Natural language generation has become an achievable task. Besides, sound scripts are of great importance to make a good movie. So, a deep learning system generating scripts for movies can save a lot of human and financial resources.

3.3 Related Works

Movie script generation is relatively new to natural language generation research. So, there are not many works on this. However, automatic story generation and plot generation works can be found. Numerous models are

there which can generate stories on provided conditions.

Plan-driven story generation has been worked on since early 2000. The objective of this problem statement is to generate a story given a plan. Some models also incorporate the possibility of other conditions [PÉrez and Sharples, 2001, Riedl and Young, 2010, Fan et al., 2019].

Although story generation and plot generation are not defined as the same objective, they can be modified to perform similar tasks. There has been notable progress in the story generation field. However, specifically for movie plot writing is yet to be explored. [Rashkin et al., 2020] paper generated stories from a given set of outline phrases. Many other story-generation works are there which can be used for similar purposes.

On the other hand, the scene generation task is not at all explored. There are works on dialogue generation such as [Li et al., 2016] and [Huang et al.,]. However, dialogue generation differs in many aspects from script generation. There are notable differences between them. A script is not just a collection of dialogues, it is properly annotated to avoid confusion between multiple speakers. Also, the annotations make place for scene description, such as location and time of the day. So collectively, these differences make the task much more disparate. Among recent works, [Zhu et al., 2022] provides impressive results on scene generation tasks. They performed scene generation from a given narrative in a retrieval-based setting. The results outperform baseline models. However, case studies show that still the models fail in a lot of trivial cases. The authors suspect due to the retrieval-based setting, the under-performance is caused.

3.4 Work at CFILT

The Computation for Indian Language Technology Lab (CFILT Lab) of IIT Bombay has been working on many cutting-edge research problems for a long time. Movie script generation is one such topic. CFILT Lab works include developing a script generator model named Kurosawa which is comprised of the mentioned two stages.

Kurosawa is a GPT-3 based model with fine-tuning on script generation-specific objectives. The work also includes creating a new dataset which is the first of its kind. The contributions are elaborated on below.

3.4.1 Dataset Generation

IIT Bombay CFILT Lab has worked on creating a completely new dataset for the said objective. For the plot generation task, plots from Wikipedia and prompts [1] and genres from IMDb are collected for the movies. IMDb has two kinds of prompts. A short description of around 15-40 words while a long storyline of about 100-200 words.

The team has created a plot generation dataset of 1000 instances of Bollywood and Hollywood movies. Each plot on average is around 700 words long. The dataset helps in further fine-tuning any pre-trained model or even training new models also.

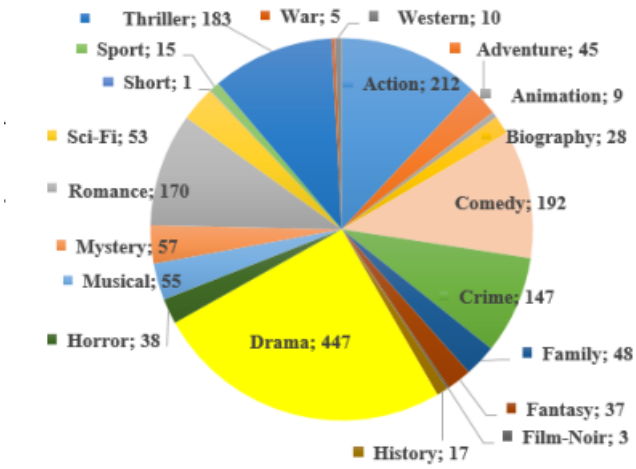


Figure 3.3: Distribution of different genres in the dataset

Figure 3.3 shows the distribution of different genres that are collected in the dataset. The scene generation data on the other hand comprises scripts

[1] A prompt for a movie is a line with the set of keywords to give an overview of the plot.

of particular short scenes. According to the collected data, a movie of around 2 hours in length contains nearly thirty thousand words in the script. The team collected movie scripts from IMSDb and annotated them manually.

The plot annotations are defined on the 4-act structure [2]. Corresponding to the 4-act structure, at the end of each act a tag is placed to mark the annotation. The four tags are *<on>, <ta>, <tb>, <th>*. For example, in figure 3.4 the plot of the movie *16 December* is shown which is also annotated with the four tags.

```
Major General Vir Vijay Singh, Vikram, Sheeba, and Victor, who are Indian Revenue Service officers
belonging to the Department of Revenue Intelligence, and have been wrongly implicated in the killing
of their corrupt superior officer and removed from service, are hired by the Chief of the same
agency to investigate a series of large money laundering. <on>
Vikram and Sheeba share some romance. The team is equipped with hi-tech equipment such as mini spy
cameras, computers, the internet and other communication devices. Through various encounters, they
discover that the money is being transferred to a Swiss Bank account. By means of an Indian
employee, Sonal Joshi working in the Auckland, New Zealand branch of the same bank, they investigate
the account in New Zealand and, with her help, find that the money is being transferred to an
international terrorist organization named Kaala Khanjar. <ta>
This organization, working in conjunction with terrorist Dost Khan, manages to smuggle a Russian-
made nuclear bomb into India. Dost Khan plans to explode the nuclear uvva on the same day, 16
December. Although the ruling dictator of Pakistan surrendered unconditionally to India during the
Indo-Pakistani War of 1971, some of the hard-lined Pakistani soldiers were bitter and angry at the
surrender, as they wanted to continue fighting the Indians until their last breath. They retreated
in silence and later formed their own groups of communal soldiers to carry out terrorists attacks
against neighboring India. Led by Dost Khan, a hardliner Pakistani army officer, who, against his
wishes, had to surrender after the end of the 1971 war, the terrorists planned to take an act of
revenge by having a nuclear explosion in the heart of New Delhi. They transport it into a music
competition disguised as a musical instrument. <tb>
When Vir Vijay Singh discovers the plan, he plans to find out the location of the nuclear bomb as
soon as possible by taking the help of Remote Radiation Sensors in satellites and innumerable
beggars in the city. This helps the team zero in on the location. After they overpower most of the
terrorists in a commando operation, Dost Khan learns about it and sets the nuclear bomb to explode
in a few minutes. This creates a lot of problem for Vijay Vir Singh, as the bomb can be defused only
by the exclusive voice command of Dost Khan saying: Dulhan Ki Vidaai Ka Waqt Badalna Hai. They adopt
a novel way to do it by speaking to Dost Khan and making him say fragments of this sentence without
making him realize that it was being done to defuse the bomb. After the conversation is over, they
synthesize the sentence to defuse the bomb just in time. <th>
```

Figure 3.4: Annotated plot of the movie 16 December

Scene annotations are done based on four different types of lines in a scene. Namely, *sluglines, action lines, dialogues* and *character names*. Any

[2] A 4-act structure consists of four parts of the story. **Act 1** is the introductory part of the act. **Act 2A** is the part where the story builds as the protagonist goes through the journey. **Act 2B** is where problems arise concerning the protagonist and in **Act 3** the conclusion comes through the climax

other form of information or metadata is removed from the data.

Sluglines	<bsl>..<esl>
Action Lines	<bal>..<eal>
Character Names	<bcn>..<ecn>
Dialogue	<bd>..<ed>

Table 3.1: Scene annotation and their tags

The annotations are marked with the mentioned tags of table 3.1. In figure **??** a portion of a movie scene with four major annotations is shown.

```
<bsl> INT. KENNY'S THAI FOOD DINER - DAY <esl>
<bal>
Kat and Mandella pick apart their pad thai.  Mandella is
smoking.
 <eal>
<bcn> KAT <ecn>
<bd> So he has this huge raging fit about
Sarah Lawrence and insists that I go to
his male-dominated, puking frat boy,
number one golf team school. I have no
say at all.
 <ed>
<bcn> MANDELLA <ecn>
<bd> William would never have gone to a
state school.
 <ed>
```

Figure 3.5: Scene Annotation

3.4.2 Experiments

GPT-3 is a large language model, published by OpenAI in 2020 [Brown et al., 2020]. It is trained with around 175 billion parameters. The team fine-tuned the GPT-3 models for plot and scene generation tasks. For plot generation, they have trained multiple models with different objectives such as, with short or long prompts or with or without genres. The details are given in table **??**

Original (O)	Non-annotated plots with short storylines
Annotation and short input (AS)	Annotated plots with short storylines
Annotation and long input (AL)	Annotated plots with long storylines
Annotation and short input with genres (ASG)	Annotated plots with short storyline and genre
Annotation and long input with genres (ALG)	Annotated plots with long storyline and genre

Table 3.2: Different Objectives of Plot Generation Training

3.4.3 Results

The metrics used to measure the efficiency of the models are the BLEU score and the ROUGE score. The scores are shown in table 3.3.

Models	BLEU-2 Score	ROUGE-L Score
O	12.95	22.67
AS	12.01	21.72
AL	13.08	24.02
ASG	12.51	23
ALG	**14.52**	**24.88**

Table 3.3: Result metric of GPT-3 fine-tuning on Plot Generation Task

The results of human evaluation suggest that the dataset must be skewed as the model generates coherent text in some cases and misses in others. The Bollywood dataset is skewed towards older movies, hence the outputs are of old themes. Also, in some cases, the text saw hallucinations.

For scene generation the scores based on *fluency*, *creativity*, *likability*, *coherence* and *relevance* are described as 4.48, 3.9, 3.48, 3.46 and 3.86. The scores are all above average and also human evaluation supports the score. The generated scenes are coherent and well-structured and adhere to the input texts.

An example of the generated plot from the model is shown in figure **??**.

Storyline:
An alien on Earth loses the only device he can use to communicate with his spaceship. His innocent nature and child-like questions force the country to evaluate the impact of religion on its people.
Plot:
An alien lands in a field in Lincoln, Nebraska, on a meteorite, and decides to stay in the area. He soon becomes interested in a nearby town, and in a library he encounters the main character, 8-year-old Owen Garvey. After drawing a spaceship on a library floor with a stick, Owen asks the alien what he is. The alien explains that he is from another planet, and that his people are in danger of being destroyed by pollution. Knowing that Earth is populated by many different religious groups, he asks Owen to be his ambassador to Earth and to help him save his people. <on>…

Figure 3.6: An example of plot generated by Kurosawa model

3.5 Other Works

As mentioned earlier, being one of the most crucial cutting-edge research, this task is still under rapid progress. [Zhu et al., 2022] paper contributed notably to the work of movie script generation. They have generated movie scripts from a given narrative, hence their work is primarily about scene generation only.

3.5.1 Dataset

The authors collected movie narratives from the GraphMovie platform where narratives are written by movie watchers. They collected on average 367 reviews of each movie and filtered them. Each such description has on average 1-3 sentences to summarize the whole movie or a fragment of the movie. The authors chose the top 100 IMDb movies for the dataset generation and annotated the data through external agencies.

The data contains nearly 16109 script sessions. A narrative has about 25 words and a session has 4.7 lines. These statistics are shown in figure 3.7.

	Training	Validation	Test
# Sessions	14,498	805	806
# Micro-sessions	136,524	37,480	38,320
# Candidates	2	10	10
Min. #lines in Session	2	2	2
Max. #lines in Session	34	27	17
Avg. #lines in Session	4.71	4.66	4.75
Avg. #words in Narrative	25.04	24.86	24.18

A narrative is a description that summarizes a fragment of a movie. Each narrative corresponds to a session containing several lines of script. Micro-sessions are obtained by moving the prediction point through the session, each of which has a sequence of previous lines at that point of time, the same narrative as the session, and the next line to predict. The line candidates are used for prediction, which contain one golden line and several lines randomly sampled from the dataset.

Figure 3.7: Dataset statistics

3.5.2 Experiment

The authors developed an attention-based architecture named ScriptWriter-CPre which performs the task. The authors designed the model keeping three specific objectives in mind. The first idea of those is an updating mechanism. The authors kept a continuous matrix to keep note of what has been said already from the narrative and therefore what is to be said next. Next is a supplementary loss, which is named Content Prediction Loss determines which content to be conveyed in the next line. And a matching score is generated from the retrieval-based [3] setting to determine cross-entropy loss. The architecture is described in figure 3.7. The image has been produced by the authors.

[3]Retrieval-based setting means here the model is not supposed to generate the next line. Rather the model will be given a few options for the best possible next line and it has to choose among them.

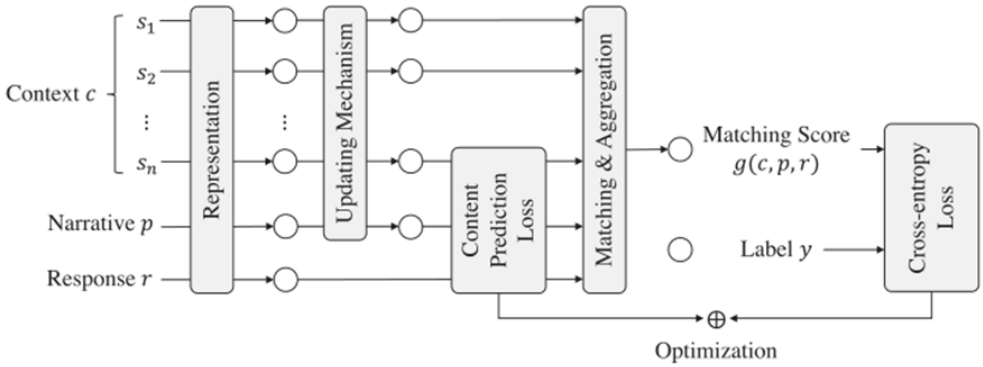


Figure 3.8: The Architecture of ScriptWriter-CPre [[Zhu et al., 2022]]

3.5.3 Results

The results were better than the other baseline models. The authors measured recall at different positions among all the candidates. $R_n@k$ refers to recall at position k among n candidates. P refers to precision. Authors measured strict and weak precision which refers to precision at exactly the right position and precision with the presence of the word.

	Turn-level					Session-level	
	$R_2@1$	$R_{10}@1$	$R_{10}@2$	$R_{10}@5$	MRR	P_{strict}	P_{weak}
MVLSTM	$0.651^{\dagger}$	$0.217^{\dagger}$	$0.384^{\dagger}$	$0.732^{\dagger}$	$0.395^{\dagger}$	$0.198^{\dagger}$	$0.224^{\dagger}$
DL2R	$0.643^{\dagger}$	$0.210^{\dagger}$	$0.321^{\dagger}$	$0.638^{\dagger}$	$0.314^{\dagger}$	$0.230^{\dagger}$	$0.243^{\dagger}$
SMN	$0.641^{\dagger}$	$0.176^{\dagger}$	$0.333^{\dagger}$	$0.696^{\dagger}$	$0.392^{\dagger}$	$0.197^{\dagger}$	$0.236^{\dagger}$
DAM	$0.631^{\dagger}$	$0.240^{\dagger}$	$0.398^{\dagger}$	$0.733^{\dagger}$	$0.408^{\dagger}$	$0.226^{\dagger}$	$0.236^{\dagger}$
DUA	$0.654^{\dagger}$	$0.237^{\dagger}$	$0.403^{\dagger}$	$0.736^{\dagger}$	$0.396^{\dagger}$	$0.223^{\dagger}$	$0.251^{\dagger}$
IMN	$0.686^{\dagger}$	$0.301^{\dagger}$	$0.450^{\dagger}$	$0.759^{\dagger}$	$0.463^{\dagger}$	$0.304^{\dagger}$	$0.325^{\dagger}$
IOI	$0.710^{\dagger}$	$0.341^{\dagger}$	$0.491^{\dagger}$	$0.774^{\dagger}$	$0.464^{\dagger}$	$0.324^{\dagger}$	$0.337^{\dagger}$
MSN	$0.724^{\dagger}$	$0.329^{\dagger}$	$0.511^{\dagger}$	$0.794^{\star}$	$0.464^{\dagger}$	$0.314^{\dagger}$	$0.346^{\dagger}$
ScriptWriter	$0.730^{\dagger}$	$0.365^{\dagger}$	0.537	0.814	0.503	0.373	$0.383^{\star}$
ScriptWriter-CPre	**0.756**	**0.398**	**0.557**	**0.817**	**0.504**	**0.392**	**0.409**

The turn-level evaluation aims at measuring the performance of models on predicting a specific line in a session, while the session-level evaluation considers the quality of the whole session. † and ⋆ denote significant differences between each baseline and ScriptWriter-CPre measured in t-test with $p \leq 0.01$ and $p \leq 0.05$, respectively.

Figure 3.9: Results of ScriptWriter-CPre by [Zhu et al., 2022]

We can see in 3.9 the results for ScriptWriter-CPre is better than the other models. The model achieved 2.6% improvement over other models. It also generates coherent and consistent text throughout.

In human evaluation also the model is chosen to be better suited for this task. ScriptWriter-CPre scored 3.5025 while MSN scored 3.3150 based on human judgement, However, there are numerous cases where the model still fails to guess the next line. One such example is shown in figure 3.10. The model hallucinates and couldn't cover the story at all here.

Narrative: Maureen swam back to the bottom of the sea, asking other fish if they saw a boat. But they didn't reply Maureen.
Ground-truth Script: (1) Do you see a boat? (2) A white boat! (3) They took my son! (4) My son! Help me! Please! (5) EOS
Generated Script: (1) Do you see a boat? (2) Do you see it? (3) Do you see it? (4) That has nothing to do with you! (5) Talk to me at least!
Error: Redundant; No ending

Figure 3.10: Case study of ScriptWriter-CPre [[Zhu et al., 2022]]

3.6 Summary

Movie script generation is one of the cutting-edge research ideas of Natural Language Generation. A lot of work is going on in this field right now which includes IIT Bombay CFILT lab work too. The contribution to generating a completely new and first-of-its-kind dataset will help in future work also. Besides, the study also shows how different teams have progressed so far and how the future work surrounds them.

Chapter 4

Data to Text Generation

4.1 Problem Statement and Motivation

The twenty-first century is a data-driven century. Everything now constitutes a huge amount of data. Most of the research across the globe is processed by collecting and observing huge amounts of data. Also, the information presented on the web constitutes a large amount of data. All of these data are structured in a specific way. For example, we can think of Wikipedia infobox, there is a certain pattern in it. The examples we discussed in Chapter 2, like medical reports, and weather forecast graphs also contain important structured data.

This data if presented in human understandable language, such as natural text, will enhance the understanding the comprehension of everyone hugely. That is the goal of the data-to-text generation task.

The data can be structured in the form of a tree or graphs, such as RDF triples. An RDF is a triple consisting of *subject*, *relation*, *object*. For example,

```
<Dumdum Airport, servesCity, Kolkata>
```

is a triple which denotes information. However, it can be converted into the following sentence.

```
Kolkata is served by the Dumdum Airport.
```

Another such example is shown in figure 4.1, which is taken from the paper [Zhao et al., 2020].

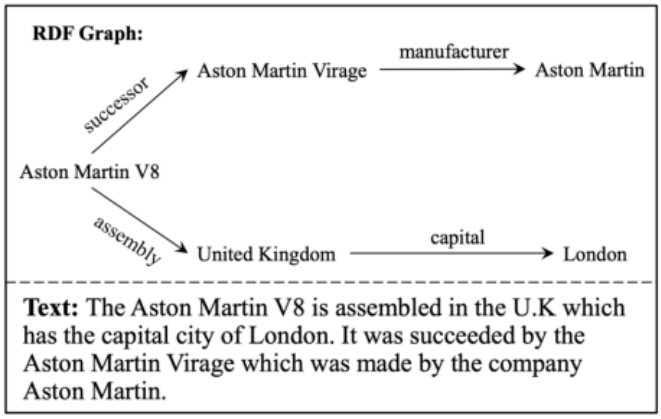
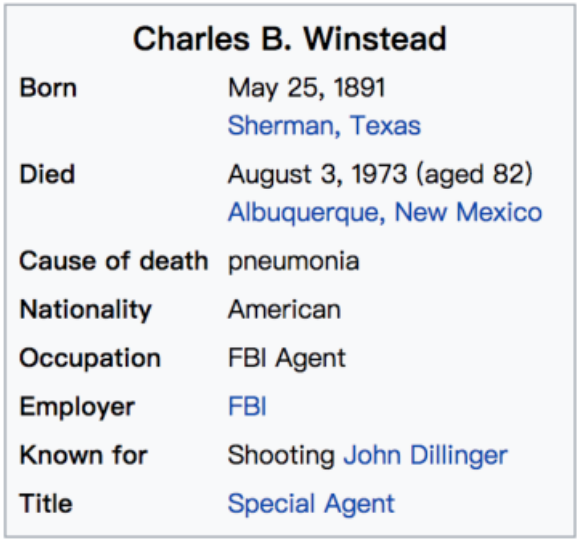


Figure 4.1: An example of textual sentence generated from an RDF

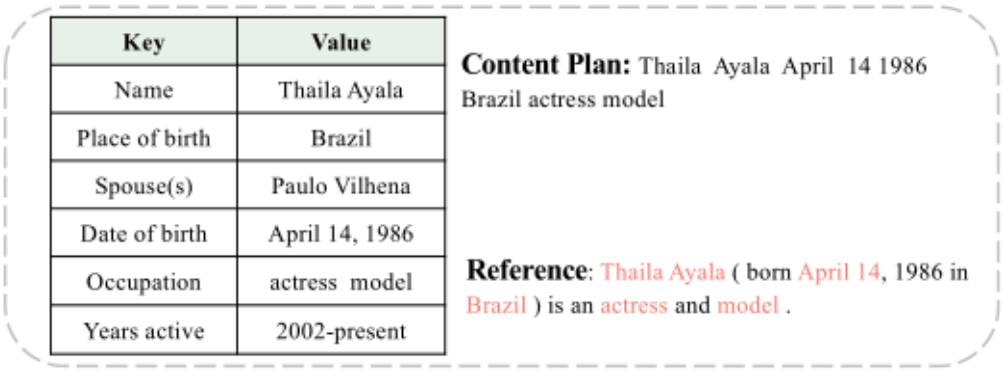
Another such data representation can be with tables. Medical reports, Wikipedia infobox (figure 4.2 from [Liu et al., 2017]) and many other data are stored in table-based format (figure 4.3 from [Zhao et al., 2023]). So, converting a table into text turns out to be another significant problem statement.



Charles B. Winstead

Born	May 25, 1891 Sherman, Texas
Died	August 3, 1973 (aged 82) Albuquerque, New Mexico
Cause of death	pneumonia
Nationality	American
Occupation	FBI Agent
Employer	FBI
Known for	Shooting John Dillinger
Title	Special Agent

Figure 4.2: Wikipedia Infobox Data



Key	Value
Name	Thaila Ayala
Place of birth	Brazil
Spouse(s)	Paulo Vilhena
Date of birth	April 14, 1986
Occupation	actress model
Years active	2002-present

Content Plan: Thaila Ayala April 14 1986 Brazil actress model

Reference: Thaila Ayala (born April 14, 1986 in Brazil) is an actress and model .

Figure 4.3: Table format data

4.2 Related Works

Data-to-text generation is one of the most growing fields of NLP. There are researchers all over the world working on these topics. Some of significant contributions in table-to-text generation are made by [Liu et al., 2017], [Zhao et al., 2023] and [Li et al., 2023]. [Liu et al., 2017] used a sequence to sequence the learner to understand the piece of information represented in

the format of a table. They modified the transformer encoder and decoder architectures to achieve the goal.

On the other hand, in RDF to text generation [Gao et al., 2020] and [Zhao et al., 2020] has shown considerable progress. Although the metrics are not absolutely human generation equivalent, however, the performance is indeed applicable to systems. [Gao et al., 2020] used a GCN encoder and Graph-based meta path encoder to encode the knowledge graph information and thus a typical transformer decoder generates the sentence.

4.3 Datasets

There are many datasets available to perform these tasks. The WebNLG Challenge dataset [Gardent et al., 2017] consists of two parts. The first is of the 2017 challenge dataset and the second is of 2020. WebNLG 2017 dataset only contains around 18102 training, 2268 validation and 2495 test instances. DART [Nan et al., 2021] is another popular dataset to perform RDF-to-text conversion tasks.

4.4 Work at CFILT

CFILT Lab of IIT Bombay is working on the problem of RDF-to-text conversion on the WebNLG challenge dataset. The model developed by the team consists of two stages. The stages are shown in figure 4.4. For the stage 1 sequence-to-sequence learner T5 model is used and fine-tuned. T5 is a pre-trained transformer-based model. It is considered to be sound for text-generation tasks.

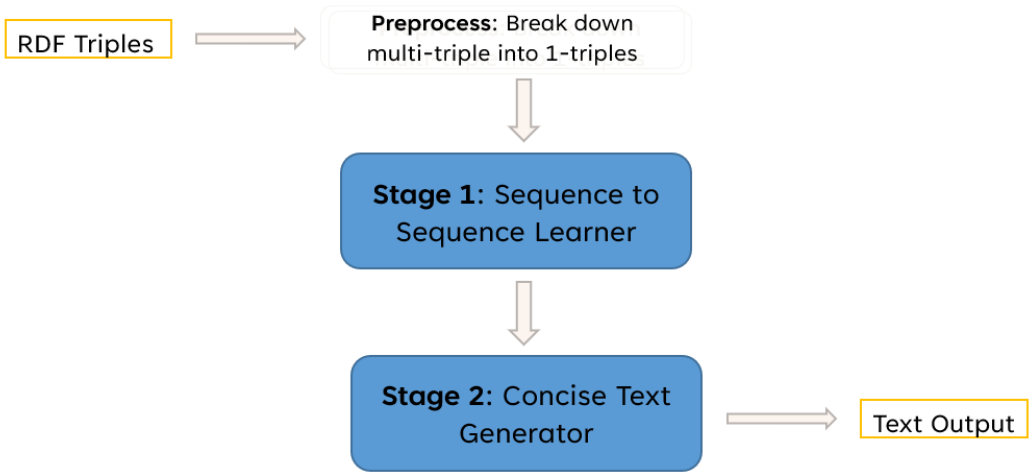


Figure 4.4: Two staged pipeline design

For the stage 2 concise text generator, prompting is to be used. Prompting on a language model or other pre-trained model often generates coherent outputs. The objective here is to generate a sentence output for each triple (which represents an edge in the graph) and then combine those sentences together into one output text.

Chapter 5

Summary, Conclusion and Future Work

5.1 Summary

Throughout the study, we have discussed several different topics. Starting from the Introduction we learnt about Natural Language Generation, why it is important in modern scenarios and about the whole study.

Then we moved on to learn about the backgrounds of NLG. NLG is a vast domain containing numerous objectives. First, we studied about prospects of an NLG system, how such a system is built and what are the challenges. Also, we talked about how NLG can be impactful in various modern applications and what exactly NLG accomplishes. We moved on to learn about the different tasks under NLG. We discussed each task, and the datasets available for the same. Therefore, we discussed what are the tools of techniques we use to solve these problems.

In Chapter 3, we studied everything about script generation, what is the problem and how exactly it can be solved. How CFILT lab is working in this domain and our contributions. Similarly, we also learnt the same about data-to-text generation in Chapter 4.

The study helped us go through the details of works in this field and comprehend them carefully. Also, we took an overview of the field at times,

which sums up the entire objective and goal of the study.

5.2 Conclusion

CFILT lab is working extensively in script generation. However, there are additional challenges to deal with. Hence, the performance of these models is yet far away from the expected objective. Many of these works are yet to be explored fully. Even from the historical perspective of scientific paradigms, we have seen how long it takes to develop a sound theory or a highly efficient system. But, these NLG problems are being dealt with for a couple of years only, or hardly a decade. Hence, we need to understand that it takes time to achieve perfection.

Creating new datasets which are unbiased, well-structured and sound takes a lot of time and effort. So, these additional issues must also be kept in mind while concluding these studies.

Although we have seen an overview of the Natural Language Generation, it is never the full picture. NLG is one of the most highly active, growing and dynamic fields of Machine Learning and whole Computer Science. Hence, an overview is never a whole overview. In this era of ChatGPT, many of the problem statements may seem to be naive in front of large language models, but it is never so.

The world of science and technology has seen a lot of such groundbreaking eras. Like the advent of calculators didn't drastically reduce the importance of mathematicians, like the invention of computers never reduced the workload for human beings, it is unlikely that large language models will reduce the necessities of NLP research. Rather from a historical perspective, we can assume that such newer problem statements have to generate from the progress.

5.3 Future Work

The future of the Natural Language Generation is bright and beautiful. The tasks we discussed in this study are one of the most growing ones. Hence, there is a long way to go indeed.

Script generation task yet to have a large-scale well structured and unbiased dataset. Also, it has a lot more to develop to achieve real-world implementable efficiency. So does apply to data-to-text generation. Besides, as we learnt from the conclusion there can be bigger objectives and goals beyond the scope of this study. So, solving them with NLG techniques does fall under the prospects too.

References

[Bajaj et al., 2018] Bajaj, P., Campos, D., Craswell, N., Deng, L., Gao, J., Liu, X., Majumder, R., McNamara, A., Mitra, B., Nguyen, T., Rosenberg, M., Song, X., Stoica, A., Tiwary, S., and Wang, T. (2018). Ms marco: A human generated machine reading comprehension dataset.

[Brown et al., 2020] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners.

[Buchanan et al., 1995] Buchanan, B. G., Moore, J. D., Forsythe, D. E., Carenini, G., Ohlsson, S., and Banks, G. (1995). An intelligent interactive system for delivering individualized information to patients. *Artificial intelligence in medicine*, 7(2):117–154.

[Caldwell and Korelsky, 1994] Caldwell, D. and Korelsky, T. (1994). Bilingual generation of job descriptions from quasi-conceptual forms. In *Proceedings of the Fourth Conference on Applied Natural Language Processing (ANLP'94)*, pages 1–6.

[Cawsey et al., 1995] Cawsey, A., Binsted, K., and Jones, R. (1995). Personalised explanations for patient education. In *Proceedings of the fifth european workshop on natural language generation*, pages 59–74.

[Dong et al., 2022] Dong, C., Li, Y., Gong, H., Chen, M., Li, J., Shen, Y., and Yang, M. (2022). A survey of natural language generation. *ACM Comput. Surv.*, 55(8).

[Fan et al., 2019] Fan, A., Lewis, M., and Dauphin, Y. (2019). Strategies for structuring story generation. *arXiv preprint arXiv:1902.01109*.

[Feng et al., 2018] Feng, X., Liu, M., Liu, J., Qin, B., Sun, Y., and Liu, T. (2018). Topic-to-essay generation with neural networks. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4078–4084. International Joint Conferences on Artificial Intelligence Organization.

[Gao et al., 2020] Gao, H., Wu, L., Hu, P., and Xu, F. (2020). Rdf-to-text generation with graph-augmented structural neural encoders. In Bessiere, C., editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3030–3036. International Joint Conferences on Artificial Intelligence Organization. Main track.

[Gardent et al., 2017] Gardent, C., Shimorina, A., Narayan, S., and Perez-Beltrachini, L. (2017). The WebNLG challenge: Generating text from RDF data. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 124–133, Santiago de Compostela, Spain. Association for Computational Linguistics.

[Goldberg et al., 1994] Goldberg, E., Driedger, N., and Kittredge, R. I. (1994). Using natural-language processing to produce weather forecasts. *IEEE Expert*, 9(2):45–53.

[Hermann et al., 2015] Hermann, K. M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., and Blunsom, P. (2015). Teaching machines to read and comprehend. In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

[Hu and Liu, 2004] Hu, M. and Liu, B. (2004). Mining and summarizing customer reviews. KDD '04, page 168–177, New York, NY, USA. Association for Computing Machinery.

[Huang et al.,] Huang, T.-H. K., Ferraro, F., Mostafazadeh, N., Misra, I., Agrawal, A., Devlin, J., Girshick, R., He, X., Kohli, P., Batra, D., Zitnick, C. L., Parikh, D., Vanderwende, L., Galley, M., and Mitchell, M. Visual storytelling. pages 1233–1239. Association for Computational Linguistics.

[Iordanskaja et al., 1992] Iordanskaja, L., Kim, M., Kittredge, R., Lavoie, B., and Polguere, A. (1992). Generation of extended bilingual statistical reports. In *COLING 1992 Volume 3: The 14th International Conference on Computational Linguistics*.

[Lai et al., 2017] Lai, G., Xie, Q., Liu, H., Yang, Y., and Hovy, E. (2017). RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark. Association for Computational Linguistics.

[Lewis et al., 2020] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

[Li et al., 2016] Li, J., Monroe, W., Ritter, A., Jurafsky, D., Galley, M., and Gao, J. (2016). Deep reinforcement learning for dialogue generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Austin, Texas. Association for Computational Linguistics.

[Li et al., 2023] Li, L., Geng, R., Fang, C., Li, B., Ma, C., Li, B., and Li, Y. (2023). Plan-then-seam: Towards efficient table-to-text generation.

[Li et al., 2017] Li, Y., Su, H., Shen, X., Li, W., Cao, Z., and Niu, S. (2017). DailyDialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 986–995, Taipei, Taiwan. Asian Federation of Natural Language Processing.

[Lin, 2004] Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

[Liu et al., 2017] Liu, T., Wang, K., Sha, L., Chang, B., and Sui, Z. (2017). Table-to-text generation by structure-aware seq2seq learning.

[McAuley and Leskovec, 2013] McAuley, J. J. and Leskovec, J. (2013). From amateurs to connoisseurs: Modeling the evolution of user expertise through online reviews. WWW '13, page 897–908, New York, NY, USA. Association for Computing Machinery.

[McKeown, 1985] McKeown, K. R. (1985). Discourse strategies for generating natural-language text. *Artificial intelligence*, 27(1):1–41.

[Mikolov et al., 2013] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Burges, C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

[Nan et al., 2021] Nan, L., Radev, D., Zhang, R., Rau, A., Sivaprasad, A., Hsieh, C., Tang, X., Vyas, A., Verma, N., Krishna, P., Liu, Y., Irwanto, N., Pan, J., Rahman, F., Zaidi, A., Mutuma, M., Tarabar, Y., Gupta, A., Yu, T., Tan, Y. C., Lin, X. V., Xiong, C., Socher, R., and Rajani, N. F. (2021). Dart: Open-domain structured data record to text generation.

[Okanda et al., 2015] Okanda, M., Asada, K., Moriguchi, Y., and Itakura, S. (2015). Understanding violations of gricean maxims in preschoolers and adults. *Frontiers in Psychology*, 6.

[Papineni et al., 2002] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (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.

[Paris et al., 1995] Paris, C., Vander Linden, K., Fischer, M., Hartley, A., Pemberton, L., Power, R., and Scott, D. (1995). A support tool for writing multilingual instructions. In *International Joint Conference on Artificial Intelligence*, volume 14, pages 1398–1404. Citeseer.

[Paulus et al., 2017] Paulus, R., Xiong, C., and Socher, R. (2017). A deep reinforced model for abstractive summarization.

[PÉrez and Sharples, 2001] PÉrez, R. P. Ý. and Sharples, M. (2001). Mexica: A computer model of a cognitive account of creative writing. *Journal of Experimental & Theoretical Artificial Intelligence*, 13(2):119–139.

[Rajpurkar et al., 2016] Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (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.

[Rashkin et al., 2020] Rashkin, H., Celikyilmaz, A., Choi, Y., and Gao, J. (2020). PlotMachines: Outline-conditioned generation with dynamic plot state tracking. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4274–4295, Online. Association for Computational Linguistics.

[Reiter and Dale, 1997] Reiter, E. and Dale, R. (1997). Building applied natural language generation systems. *Natural Language Engineering*, 3(1):57–87.

[Reiter et al., 1995] Reiter, E., Mellish, C., and Levine, J. (1995). Automatic generation of technical documentation. *Applied Artificial Intelligence an International Journal*, 9(3):259–287.

[Riedl and Young, 2010] Riedl, M. O. and Young, R. M. (2010). Narrative planning: Balancing plot and character. *Journal of Artificial Intelligence Research*, 39:217–268.

[Rush et al., 2015] Rush, A. M., Chopra, S., and Weston, J. (2015). 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. Association for Computational Linguistics.

[Safovich and Azaria, 2020] Safovich, Y. and Azaria, A. (2020). Fiction sentence expansion and enhancement via focused objective and novelty curve sampling. In *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 835–843.

[Socher et al., 2013] Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., and Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

[Song et al., 2019] Song, K., Tan, X., Qin, T., Lu, J., and Liu, T.-Y. (2019). MASS: Masked sequence to sequence pre-training for language generation. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5926–5936. PMLR.

[Tang et al., 2017] Tang, J., Wang, Y., Zheng, K., and Mei, Q. (2017). End-to-end learning for short text expansion. KDD '17, page 1105–1113, New York, NY, USA. Association for Computing Machinery.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

[Zhang et al., 2020] Zhang, J., Zhao, Y., Saleh, M., and Liu, P. (2020). PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339. PMLR.

[Zhang et al., 2018] Zhang, S., Dinan, E., Urbanek, J., Szlam, A., Kiela, D., and Weston, J. (2018). Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213, Melbourne, Australia. Association for Computational Linguistics.

[Zhao et al., 2020] Zhao, C., Walker, M., and Chaturvedi, S. (2020). Bridging the structural gap between encoding and decoding for data-to-text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2481–2491, Online. Association for Computational Linguistics.

[Zhao et al., 2023] Zhao, Y., Qi, Z., Nan, L., Flores, L. J. Y., and Radev, D. (2023). Loft: Enhancing faithfulness and diversity for table-to-text generation via logic form control.

[Zhu et al., 2022] Zhu, Y., Song, R., Nie, J.-Y., Du, P., Dou, Z., and Zhou, J. (2022). Leveraging narrative to generate movie script. *ACM Trans. Inf. Syst.*, 40(4).