GENERATING A NARRATIVE BY GIVING PROMPTS TO PRE-TRAINED MODELS WITH EXCELLENT STORY TELLING CAPABILITIES

MOHAMMED AAMER   
(Student ID: PN1130101)

Final Thesis Report

September 2024

# **DEDICATION**

# To my family, whose unwavering support and encouragement have been my foundation throughout this journey. Your love, patience, and belief in my dreams have been my greatest motivation.

To my advisor Dr. Abdel Karim Tamimi, for your invaluable guidance, insightful feedback, and constant encouragement. Your mentorship has been instrumental in shaping this thesis and my academic growth.

And to my friends and colleagues, who provided camaraderie and inspiration along the way. Your friendship has made this challenging journey a rewarding experience.

Thank you all for being a part of this achievement.

# **ACKNOWLEDGEMENTS**

I extend my gratitude to Dr. Abdel Karim Tamimi for his invaluable guidance and counsel throughout the course of this thesis. I am profoundly grateful to my parents for their unwavering support, without which this endeavor would not have been achievable. I also express my appreciation to my fellow students for their encouragement and assistance. Finally, I acknowledge Upgrad and LJMU for providing the opportunity to undertake this research.

# **ABSTRACT**

The exploration of Automatic Story Generation has been diverse, but the advent of Large Language Models has added a new aspect by leveraging Prompt-Learning to create coherent text in Zero-Shot and Few-Shot contexts. While Large Language Models can produce coherent text, challenges persist in areas like Controllable Story Generation, which requires a consistent flow of events.

This thesis addresses this challenge by reviewing existing studies on Automatic Story Generation and Prompt-Learning. It explores the principles of In-Context Learning and Instruction Tuning within LLMs. Evaluation comprises of various criteria, inclusive of the size and type of Language Models, employing diverse metrics for evaluation both quantitative & qualitative parameters. Through an analysis of successes and failures, the thesis offers insights into potential avenues for future improvement.

# **TABLE OF CONTENTS**

DEDICATION i

ACKNOWLEDGEMENTS ii

ABSTRACT iii

[LIST OF TABLES](#_Toc128424742) iv

[LIST OF FIGURES](#_Toc128424743) v

[LIST OF ABBREVIATIONS](#_Toc128424744) vi

[CHAPTER 1: INTRODUCTION 1](#_Toc128424745)5

[1.1 Context of the Research 15](#_Toc128424746)

[1.2 Research Questions 16](#_Toc128424747)

[1.3 Goals and Objectives 16](#_Toc128424748)

[1.4 Importance of the Research 17](#_Toc128424749)

[1.5 Scope of the Research 17](#_Toc128424750)

[1.6 Organization of the Study 17](#_Toc128424751)

[CHAPTER 2: LITERATURE REVIEW 18](#_Toc128424752)

[2.1 Overview 18](#_Toc128424753)

[2.2 Automatic Story Generation 18](#_Toc128424754)

[2.2.1 Overview 18](#_Toc128424755)

[2.2.2 Structural Models 19](#_Toc128424756)

[2.2.3 Planning-Based Models 19](#_Toc128424757)

[2.2.4 Machine Learning Models 19](#_Toc128424758)

[2.2.4.1 Story Abstraction 20](#_Toc128424759)

[2.2.4.2 Script Learning and Generation 21](#_Toc128424760)

[2.2.4.3 Story Completion 22](#_Toc128424761)

[2.2.4.4 Story Generation 22](#_Toc128424762)

[2.2.5 Story Evaluation 2](#_Toc128424763)3

[2.2.6 Challenges in Automatic Story Generation 25](#_Toc128424764)

[2.3 Prompt-Based Learning 27](#_Toc128424765)

[2.3.1 Evolution of Prompting in NLP 27](#_Toc128424766)

[2.3.2 Introduction to Prompting 28](#_Toc128424767)

[2.3.2.1 Adding Prompts 28](#_Toc128424768)

[2.3.2.2 Search for Answers 28](#_Toc128424769)

[2.3.2.3 Mapping Answers 29](#_Toc128424770)

[2.3.2.4 Design Considerations for Prompting 29](#_Toc128424771)

[2.3.3 Pre-trained Language Models 29](#_Toc128424772)

[2.3.3.1 Training Objectives 30](#_Toc128424773)

[2.3.3.2 Noising Techniques 30](#_Toc128424774)

[2.3.3.3 Representation Directionality 31](#_Toc128424775)

[2.3.3.4 Common Pre-training Approaches 31](#_Toc128424776)

[2.3.4 Prompt Engineering 31](#_Toc128424777)

[2.3.4.1 Prompt Structure 32](#_Toc128424778)

[2.3.4.2 Manual Template Design 32](#_Toc128424779)

[2.3.4.3 Automated Template Design 32](#_Toc128424780)

[2.3.5 Answer Engineering 33](#_Toc128424781)

[2.3.5.1 Answer Format 33](#_Toc128424782)

[2.3.5.2 Methods for Answer Format Design 33](#_Toc128424783)

[2.3.6 Training Approaches for Prompting 34](#_Toc128424784)

[2.3.6.1 Training Configuration 35](#_Toc128424785)

[2.3.6.2 Methods for Parameter Updates 35](#_Toc128424786)

[2.3.7 Prompt-Related Topics 37](#_Toc128424787)

[2.3.7.1 Few-shot Learning 37](#_Toc128424788)

[2.3.7.2 QA-based Task Formulation 38](#_Toc128424789)

[2.3.7.3 Controlled Text Generation 38](#_Toc128424790)

[2.3.7.4 Data Augmentation Techniques 39](#_Toc128424791)

[2.3.8 Challenges in Prompting 39](#_Toc128424792)

[2.3.8.1 Designing Effective Prompts 39](#_Toc128424793)

[2.3.8.2 Engineering Answer Formats 40](#_Toc128424794)

[2.3.8.3 Choosing Tuning Strategies 40](#_Toc128424795)

[2.3.8.4 Learning with Multiple Prompts 41](#_Toc128424796)

[2.3.8.5 Selecting Pre-trained Models 41](#_Toc128424797)

[2.3.8.6 Prompt Transferability 42](#_Toc128424798)

[2.3.8.7 Combining Different Paradigms 42](#_Toc128424799)

[2.3.8.8 Calibrating Prompting Methods 42](#_Toc128424800)

[2.4 Summary 43](#_Toc128424801)

[CHAPTER 3: RESEARCH METHODOLOGY 44](#_Toc128424802)

[3.1 Overview 44](#_Toc128424803)

[3.2 Algorithms and Techniques 44](#_Toc128424804)

[3.2.1 Pre-Trained Language Models 44](#_Toc128424805)

[3.2.1.1 OPT 45](#_Toc128424806)

[3.2.1.2 OPT-IML 45](#_Toc128424807)

[3.2.1.3 LaMDA & PaLM 45](#_Toc128424808)

[3.2.1.4 GPT-2 46](#_Toc128424809)

[3.2.1.5 GPT-3 46](#_Toc128424810)

[3.2.1.6 GPT-3.5 47](#_Toc128424811)

[3.2.1.7 BLOOM 47](#_Toc128424812)

[3.2.1.8 BLOOMZ 47](#_Toc128424813)

[3.2.2 n-Shot Learning 48](#_Toc128424814)

[3.3 Methodology 48](#_Toc128424815)

[3.3.1 Comprehensive Pipeline 48](#_Toc128424816)

[3.3.2 Description of Data 49](#_Toc128424817)

[3.3.3 Data Preparation Process 50](#_Toc128424818)

[3.3.3.1 ROCStories Dataset 50](#_Toc128424819)

[3.3.3.2 WritingPrompts Dataset 51](#_Toc128424820)

[3.3.4 Implementation Details 51](#_Toc128424821)

[3.3.4.1 Creating Prompts from Outlines 51](#_Toc128424822)

[3.3.4.2 Generating Stories from Prompts 52](#_Toc128424823)

[3.3.5 Evaluation 53](#_Toc128424824)

[3.4 Summary 53](#_Toc128424828)

[CHAPTER 4: IMPLEMENTATION 54](#_Toc128424829)

[4.1 Overview 54](#_Toc128424830)

[4.2 Data Preparation 54](#_Toc128424831)

[4.3 Story Generation Process 56](#_Toc128424832)

[4.3.1 Procedures for Generating Stories 56](#_Toc128424833)

[4.3.2 Zero-Shot Story Generation Techniques 57](#_Toc128424834)

[4.3.3 One-Shot and Few-Shot Story Generation Techniques 57](#_Toc128424835)

[4.4 Summary 58](#_Toc128424836)

[CHAPTER 5: RESULTS & EVALUATION 59](#_Toc128424837)

[5.1 Overview 59](#_Toc128424838)

[5.2 Quantitative Analysis 59](#_Toc128424839)

[5.3 Qualitative Analysis 61](#_Toc128424840)

[5.3.1 Example of Generated Story – 1 61](#_Toc128424841)

[5.3.2 Example of Generated Story – 2 65](#_Toc128424842)

[5.4 Summary 69](#_Toc128424843)

[CHAPTER 6: CONCLUSION & RECOMMENDATIONS 70](#_Toc128424844)

[6.1 Discussion and Final Thoughts 70](#_Toc128424845)

[6.1.1 Optimal Model for Story Generation 70](#_Toc128424846)

[6.1.2 Metrics for Evaluating Story Generation 70](#_Toc128424847)

[6.2 Contribution 71](#_Toc128424848)

[6.3 Future Directions 71](#_Toc128424849)

[6.3.1 Comparison with Other Models 71](#_Toc128424850)

[6.3.2 Comparison with Existing Methods 72](#_Toc128424851)

[6.3.3 Improved Evaluation Metrics 72](#_Toc128424852)

[REFERENCES 73](#_Toc128424853)

[APPENDIX A: RESEARCH PROPOSAL 91](#_Toc128424854)

# **LIST OF TABLES**

Table 5.2.1: Evaluation Scores…………………………………………………………………...59

# **LIST OF FIGURES**

Figure 3.2.1.1: GPT-3 Model Sizes………………………………………………………………46

Figure 3.3.1.1: E2E Pipeline……………………………………………………………………...48

Figure 3.3.3.1: ROCStories Outline Generation………………………………………………….50

Figure 3.3.3.2: WritingPrompts Outline Generation……………………………………………..51

Figure 3.3.4.1: Zero-Shot Prompt………………………………………………………………...51

Figure 3.3.4.2: Few-Shot Prompt……………………………………………………………...…52

Figure 3.3.4.3; Prompting for Story Generation……………………………………………….…52

Figure 4.2.1: Data Preparation Steps……………………………………………………………..53

Figure 4.2.2: Sample Outline………………………………………………………………..……54

Figure 4.2.3: Sample Prompt……………………………………………………………………..54

Figure 4.3.1.1: Story Generation Steps……………………………………………………...……55

Figure 5.2.1: Evaluation Scores Plot……………………………………………………………..60

Figure 5.3.1.1 Outline – 1……………………………………………………………………...…61

Figure 5.3.1.2 Generated Story: GPT2 – 1………………………………………….……………62

Figure 5.3.1.3 Generated Story: Bloomz-1B7 – 1……………………………………..…………62

Figure 5.3.1.4 Generated Story: Ada – 1…………………………………………………………63

Figure 5.3.1.5 Generated Story: Babbage – 1…………………………………………………….63

Figure 5.3.1.6 Generated Story: Curie – …………………………………………………………63

Figure 5.3.1.7 Generated Story: Davinci – 1……………………………………………..........…64

Figure 5.3.2.1 Outline – 2………………………………………………...………………………65

Figure 5.3.2.2 Generated Story: GPT2 – 2……………………………………………………….66

Figure 5.3.2.3 Generated Story: Bloomz-1B7 – 2………………………………………………..66

Figure 5.3.2.4 Generated Story: Ada – 2…………………………………………………………67

Figure 5.3.2.5 Generated Story: Babbage – 2………………………………………………….…67

Figure 5.3.2.6 Generated Story: Curie – 2………………………………………………………..67

Figure 5.3.2.7 Generated Story: Davinci – 2……………………………………………..………68

# **LIST OF ABBREVIATIONS**

AI Artificial Intelligence

API Application Programming Interface

ASG Automatic Story Generation

BERT Bidirectional Encoder Representations from Transformers

BLEU BiLingual Evaluation Understudy

BLOOM BigScience Large Open-science Open-access Multilingual Language Model

CBT Children’s Book Test

CC Common Crawl

CIDEr Consensus-based Image Description Evaluation

CLM Causal Language Modeling

CTR Corrupted Text Reconstruction

EACS Embedding Average Cosine Similarity

FLAN Finetuned Language Models

FTR Full Text Reconstruction

GAN Generative Adversarial Network

GENCI Grand Équipement National de Calcul Intensif

GMS Greedy Matching Score

GPT Generative Pre-trained Transformer

GPU Graphics Processing Unit

HANNA Human-ANnotated NArratives

HTML Hyper Text Markup Language

IDRIS Institute for Development and Resources in Intensive Scientific Computing

IML Instruction Meta-Learning

L2R Left-to-Right

LAMA LAnguage Model Analysis

LaMDA Language Model for Dialogue Applications

LLM Large Language Model

LM Language Model

LSTM Long Short-Term Memory

MCNC Multiple Choice Narrative Cloze

METEOR Metric for Evaluation of Translation with Explicit ORdering

ML Machine Learning

MLM Masked Language Model

# **CHAPTER 1**

# **INTRODUCTION**

## **1.1 Context of the Research**

Emergence of Pre-trained Language Models (PLMs) has significantly enhanced the quality of machine-generated text, rendering it nearly indistinguishable from human-written text. Despite this advancement, controlling the generation process remains challenging.

Stories generated by large language models (LLMs) can have several drawbacks. They often lack deep originality, producing content that may feel formulaic or repetitive due to reliance on patterns from training data. The coherence and consistency of plot and character development can sometimes be weak, leading to confusing or disjointed narratives. Additionally, LLM-generated stories may not fully capture nuanced human emotions or cultural contexts, resulting in shallow or inappropriate themes. There’s also a risk of perpetuating biases present in the training data. Finally, the generated content might lack a personal touch, making it less engaging for readers seeking unique or heartfelt storytelling.

Prompt-based learning has emerged as a promising approach to address linguistic issues without the need for fine-tuning. By using task-specific prompts, PLMs can effectively tackle existing or new generation tasks. Prompt-based learning offers numerous advantages, such as enhancing focus and structure by directing learners towards specific topics. It fosters critical thinking and active engagement, encouraging deeper exploration and understanding. Through creative prompts, learners can explore different perspectives and solutions, promoting innovative thinking. This approach supports self-directed learning, allowing individuals to take charge of their educational journey. Additionally, prompts facilitate assessment and reflection, helping learners track their progress and identify areas for improvement. Overall, prompt-based learning makes the learning experience more dynamic and personalized.

To address the difficulties of fine-tuning, this study investigates employing prompt-based learning for generating stories and using in-context examples to guide the generation process.

## **1.2 Research Questions**

In this thesis, we will be addressing the following:

## Can the techniques for story generation that require fine-tuning pre-trained language models (PLMs) be adapted to use prompt-based learning for Few-Shot story generation, eliminating the need for fine-tuning?

## Given that previous methods primarily use GPT-2 as the base model, could employing the latest GPT-3 or other advanced models improve text generation capabilities?

## Since prompt-based learning has proven effective for text generation in zero-shot and few-shot settings, can this approach be applied to story generation to produce coherent narratives without extensive training or fine-tuning?

## **1.3 Goals and Objectives**

This study seeks to investigate prompt-based learning and assess the potential of various models for creating controlled narratives.

Objectives:

* Perform a comprehensive study of existing research.
* Assess the practicality and develop a procedure for story creation using prompt-based techniques.
* Evaluate the produced stories using automated metrics for story generation.
* Benchmark the developed approach against leading models and methodologies in the field.

## **1.4 Importance of the Research**

With short-form story generation receiving considerable attention while long-form story generation remains relatively unexplored. While previous research has largely depended on fine-tuning approaches, there is a significant gap in exploring story generation methods that do not require tuning.

## Practically, this research supports story writers by enhancing their storytelling skills with AI assistance. It aims to generate new ideas and help overcome writer’s block, thereby facilitating the creation of improved stories through collaboration with artificial intelligence.

## **1.5 Scope of the Research**

The scope is outlined below:

* Finalization of the thesis within the given timeframe post submission of the research proposal.
* Experimentation utilizing software and other resources available at our disposal.
* Utilization of the GPU resources available on a personal workstation.
* This evaluation will solely rely on metrics and exclude human assessment.

## **1.6 Organization of the Study**

Structure of this work is organized as mentioned below:

* Chapter 1 – Introduction
* Chapter 2 – Literature Review
* Chapter 3 – Methodology
* Chapter 4 – Implementation
* Chapter 5 – Results
* Chapter 6 – Conclusion

# **CHAPTER 2**

# **LITERATURE REVIEW**

## **2.1 Overview**

This chapter presents a comprehensive review of existing research in the domains of automatic story generation and prompt-based learning. It explores the evolution and advancements in these fields, highlighting key methodologies, technologies, and findings from earlier studies. The review aims to contextualize the current research within the broader landscape, identifying gaps and opportunities for further investigation. By examining both historical and contemporary approaches, this chapter sets the stage for understanding how these fields have developed and where future research might focus.

## **2.2 Automatic Story Generation**

### **2.2.1 Overview**

For a computer system to display true creativity, it must produce stories that are distinct from previous examples. This requires taking into account various elements such as the setting, the characters' goals and motivations, and their interactions and conflicts. The vast range of possible attributes in a story can make it challenging to efficiently search for and generate novel stories.

Gervás (2009) explored how these systems aim to replicate human creativity. Kybartas and Bidarra (2017) classified story generation systems into four categories: manual author writing, automated plot creation, automated space generation, and integrated story generation, which combines both automated plot and space creation methods.

### **2.2.2 Structural Models**

### There are two main methods for generating stories using computer programs: story graphs and schemas. In the story graph method, a branching graph is constructed to represent all possible story paths. A specific path is then selected to build the narrative, with the overall quality of the story graph influencing the final story's quality. An example of this technique is SCHEHERAZADE, developed by Li et al. (2013).

Although structural models are beneficial for producing well-organized stories and are relatively straightforward to implement, they have limitations. They may struggle with narratives involving multiple protagonists and can produce stories that lack coherence or believability due to insufficient attention to story semantics.

### **2.2.3 Planning-Based Models**

Critics have pointed out that story grammar theories fall short in capturing the full depth of narratives. Black and Wilensky (1979) and Black and Bower (1980) argue that these theories tend to focus more on structural elements rather than deeper narrative meanings and struggle with stories involving conflicting goals or multiple characters.

To address these limitations, Wilensky (1983) proposed the story points theory, which views a story as a sequence of causally connected events aimed at achieving a specific objective. Cook’s \*Plotto: The Master Book of All Plots\* (2011) compiles plot elements and guidelines based on this theory and has been used for creating computational narratives. Nevertheless, the Plotter system developed by Eger et al. (2015), which utilizes Plotto fragments to generate plots, encounters issues with narrative coherence because it concentrates solely on the present state of the story without considering information from previous stages.

### **2.2.4 Machine Learning Models**

Machine learning models, particularly advanced language models like GPT-3 and GPT-4, are revolutionizing the art of story creation. These models are trained on extensive text corpora and use their deep understanding of language patterns to generate coherent narratives based on initial prompts or outlines. Prompt-based generation allows users to input specific cues or partial storylines, guiding the model to produce complete narratives. Fine-tuning these models on specialized datasets enhances their ability to create stories in particular genres or styles. Controlled generation techniques, such as reinforcement learning and rule-based adjustments, help direct the model's output to meet certain thematic or stylistic requirements. Additionally, interactive story generation enables ongoing user input, creating more dynamic and personalized stories. The effectiveness of these generated stories is evaluated through metrics like perplexity and human assessments, ensuring the narratives are engaging and relevant.

#### **2.2.4.1 Story Abstraction**

#### Machine learning techniques are increasingly utilized for story abstraction, which involves summarizing or condensing a narrative into a more concise form while preserving its core elements. This process typically involves several key methods:

#### Extractive summarization: Machine learning models identify and extract significant sentences or passages from the original story to create a summary. Techniques such as sequence-to-sequence models and attention mechanisms are employed to select the most relevant content.

#### Abstractive summarization: Unlike extractive methods, abstractive summarization models generate new sentences that capture the core ideas of the story, potentially using different phrasing from the original text. This approach often relies on advanced transformer models like BERT and GPT, which have been fine-tuned on summarization tasks.

#### Text classification and clustering: Machine learning algorithms categorize parts of the story into thematic or narrative elements. This categorization helps in structuring the abstracted summary by grouping related content and reducing redundancy.

#### Reinforcement learning: Models can be fine-tuned using reinforcement learning techniques where the objective is to maximize the quality of the abstract based on predefined metrics or human feedback. This iterative process improves the abstract’s coherence and relevance.

#### Natural language understanding (NLU): Models with strong NLU capabilities can grasp the underlying context and main events of a story, allowing them to generate more meaningful and accurate abstracts.

#### By applying these machine learning methods, systems can effectively condense complex narratives into succinct summaries that capture the essence of the original text, making story abstraction both efficient and insightful.

#### **2.2.4.2 Script Learning and Generation**

Script learning and generation tasks leverage statistical models to analyze event relationships in stories, aiming to predict subsequent events based on prior sequences.

Chambers and Jurafsky (2008) utilized coreference relationships and Pointwise Mutual Information (PMI) to identify event chains and character interactions, classifying event relationships to generate a list of probable subsequent events.

Jans et al. (2012) enhanced this method by linking events to following events in the sequence and applying skip n-grams to compute statistics across the event chain. This approach addressed data sparsity and improved training by ranking events.

Script learning and generation using machine learning models involve training algorithms to create coherent and contextually relevant scripts or narratives. These models, often based on neural networks and transformer architectures, learn patterns from large datasets of existing scripts to understand dialogue, plot structure, and character interactions. By leveraging techniques such as sequence-to-sequence learning and attention mechanisms, these models can generate new scripts that follow similar styles and structures. Reinforcement learning can further refine these scripts by optimizing for specific goals like engaging dialogue or thematic consistency. Overall, machine learning models enhance script generation by automating the creation of complex and diverse narratives.

Pichotta and Mooney (2014) proposed a model that integrates multi-arguments with event structures to illustrate connections between events and entities, creating a continuous event chain throughout the story. This method demonstrated better prediction accuracy compared to approaches relying on verb-dependency pairs.

#### **2.2.4.3 Story Completion**

It focuses on enabling machines to complete a story plot based on the given context (Guan et al., 2019). Roemmele et al. (2017b) employed the Children’s Book Test (CBT) dataset for generating the complete sentence word by word, evaluating the results using various linguistic metrics to compare the generated sentence with the original.

Guan et al. (2019) created a neural network model designed to generate story endings by combining plot coherence with an understanding of the story's background knowledge, utilizing story coherence elements like events, attributes, and relationships, and incorporating ConceptNet with multi-source attention for better integration.

Wang et al. (2020) designed a model based on GPT-2 that predicts missing sentences in a story by analyzing surrounding sentences.

#### **2.2.4.4 Story Generation**

Jain et al. created a method that integrated a Statistical Machine Translation (SMT) technique for handling phrases within sentences and a deep Recurrent Neural Network (RNN) for generating complete stories from entire sentences. Despite this dual approach, the evaluation revealed poor performance.

Choi et al. (2016) used a Recurrent Neural Network (RNN) model to forecast the next sentence based on preceding ones. They employed two RNN models: an RNN Encoder-Decoder to transform sentences into embedding vectors and then back into sentences, and an RNN Story Generator to create new sentences from these vectors. While the generated sentences were grammatically accurate and contextually appropriate, they sometimes did not align well with the overall narrative.

Ammanabrolu et al. (2019) used policy gradient deep reinforcement learning to improve event generation, enhancing story quality and engagement through training four distinct event-to-text models.

Despite these advancements, RNNs have struggled with issues such as short memory and coherence. Fan et al. (2018) addressed these challenges by dividing the story generation process into two phases, using a CNN-based language model for generating story premises followed by a Seq2Seq model to turn these premises into text.

Chen et al. (2021) created story outlines before generating full stories, resulting in better coherence, though some issues persisted.

Zhai et al. (2019) employed a hybrid model combining a temporal script graph with an agenda generator to produce stories from a small corpus.

Araz (2020) used a transformer-based neural network for story generation from prompts but faced issues with deviations from prompts, grammatical errors, and repetitive content.

Pre-trained Language Models (PLMs) like GPT-2 have been investigated for story generation, but their effectiveness remains debated.

### **2.2.5 Story Evaluation**

Evaluating the quality of automatically generated stories is challenging due to the subjective nature of storytelling, the variability in evaluation criteria, and the complexity of story elements. Many current evaluation methods rely on human judgment, which can be inflexible, time-consuming, subjective, and lacking in standardized comparison criteria. Additionally, human evaluators may use their own imagination to fill in gaps, potentially introducing biases and assigning higher scores to inconsistent stories.

There are several criteria to evaluate the output of systems which generate automated stories:

* Narrative Cloze Test
* Story Cloze Test (SCT)
* Multiple Choice Narrative Cloze (MCNC) Test

We can apply various metrics for assessing the performance of the story generator. They may include coherence, fluency, novelty, engagement, and adherence to story structure. Automated evaluation metrics such as BLEU, ROUGE, METEOR, and perplexity can provide quantitative measures of the generated stories' quality. However, it is essential to complement automated metrics with human evaluation to ensure a comprehensive assessment of the generated stories.

Statistical models: Evaluation metrics for statistical models predicting story events typically include:

* + N-gram overlap
  + Perplexity
  + Pointwise Mutual Information (PMI)
* Embeddings models: Evaluation metrics for embedding models predicting story events may include:
  + Cosine similarity metrics
  + Greedy Matching Score
  + Average Maximum Similarity
  + Deep Structured Semantic model
  + Conditional Generative Adversarial Networks model

Evaluating story generation systems involves a diverse set of approaches, including linguistic analysis, statistical measures, cognitive theories, social media metrics, and assessments of suspense and interestingness. Each approach offers unique insights into the quality and effectiveness of generated stories.

Here's a breakdown of some common evaluation methods:

* Linguistic Evaluation: This involves analyzing the linguistic properties of the generated stories, such as lexical cohesion, style matching, entity co-reference, grammatical correctness, sequence of events, and narrative structure. These measures help assess the fluency, coherence, and structure of the stories.
* Statistical Measures: Customized statistical evaluation measures, such as believability scores calculated based on the believability of individual story actions, can provide quantitative assessments of story quality.
* Cognitive Theories: Some evaluation methods are based on cognitive theories of story comprehension and engagement. For example, assessing the interestingness of stories based on their ability to generate unexpected events or measuring suspense levels using methods like calculating the cost of escape plans at different time-slices.
* Social Media Metrics: Metrics like upvotes from social media posts or crowdsourced evaluations of interestingness can provide indirect indicators of story quality and engagement.
* Benchmark Datasets: Creating standardized benchmark datasets, like OpenMEVA and HANNA, can help evaluate story generation systems in a more comprehensive and standardized manner.

Despite these approaches, evaluating generated stories remains a complex and multifaceted task due to the subjective nature of storytelling and the lack of a single "correct" answer. Researchers continue to explore new methods and datasets to improve the evaluation of story generation systems and better understand their capabilities and limitations.

### **2.2.6 Challenges in Automatic Story Generation**

The challenges and limitations facing automatic story generation highlight the complexity of this field and the need for continued research and development. Let's address each of these challenges:

* Dispersion: Standardizing domain knowledge and evaluation criteria would indeed enhance the comparability of different story generation systems. By establishing common benchmarks and metrics, researchers can more accurately assess the performance of various models and identify areas for improvement. This standardization could also promote collaboration and knowledge sharing within the research community.
* Domain Knowledge: Open-domain story generation systems offer exciting opportunities to create diverse and adaptable storytelling AI. However, integrating multiple sources of knowledge, including commonsense reasoning and semantic relations, poses challenges related to data quality, standardization, and model scalability. Addressing these challenges could unlock the full potential of open-domain story generation.
* Pre-trained Language Models (PLMs): While PLMs like GPT-3 have shown impressive capabilities in generating text, they still face challenges such as repetition, coherence issues, and lack of commonsense reasoning. Further advancements in PLMs, coupled with innovative techniques to address these limitations, could significantly enhance the quality of generated stories.
* Story Interestingness: Balancing consistency, structure, and interestingness is crucial for creating engaging stories. Hierarchical models show promise in this regard by generating storylines first and then fleshing out the narrative. Incorporating insights from cognitive science and literature could further enhance the appeal of generated stories, but more research is needed to develop robust models that effectively capture these elements.
* Objective Evaluation: Developing evaluation metrics that accurately capture the quality and creativity of generated stories remains a significant challenge. Current metrics like BLEU and ROUGE are not well-suited for assessing the nuanced aspects of storytelling, and there is a need for novel evaluation methods that align better with human judgments. Collaborative efforts to define and refine evaluation criteria could lead to more meaningful assessments of story generation systems.

Addressing these challenges will require interdisciplinary collaboration, innovative research methodologies, and ongoing dialogue within the research community. By overcoming these limitations, automatic story generation could reach new heights of creativity and utility, opening up exciting possibilities for applications in entertainment, education, and beyond.

**2.3 Prompt Based Learning**

### **2.3.1 Evolution of Prompting in NLP**

The progression of paradigms in Natural Language Processing (NLP) mirrors the field's advancement towards more efficient and proficient models:

* Initial Supervised Learning Approach: At the outset, NLP models operated on fully supervised learning, where they were trained exclusively for a specific task using labeled data. This paradigm heavily relied on feature engineering, as models were constructed based on manually crafted features extracted from the data.
* Architectural Design Paradigm: This paradigm focused on devising neural network architectures with particular inductive biases to enhance performance on NLP tasks.
* Pre-training and Fine-tuning Paradigm: Towards the latter part of the 2010s, language Models (LMs) with fixed architectures, such as BERT and GPT, underwent pre-training on extensive text corpora using unsupervised learning.
* Pre-train, Prompt, and Predict Paradigm: In recent times, the pre-train, prompt, and predict paradigm has gained prominence in NLP. This approach redefines downstream tasks into prompts that the LM can interpret, eliminating the need for task-specific fine-tuning of pre-trained LMs. By supplying textual prompts, a single pre-trained LM can tackle a diverse range of tasks, albeit effective prompt formulation is vital for optimal performance.

Each paradigm signifies a shift in the methodology of training and application of NLP models, emphasizing the utilization of extensive text corpora, refinement of architectures, and simplification of downstream task formulation. The evolution of these paradigms underscores the ongoing endeavor to develop more adaptable, proficient, and broadly applicable NLP models.

### **2.3.2 Introduction to Prompting**

Prompting in machine learning, particularly in natural language processing (NLP), involves providing a model with an initial input or cue to guide its response or output. This input, known as a "prompt," helps the model generate relevant text, perform tasks, or provide answers based on the given context. This enables the LM to generate anticipated outcomes without requiring extensive labeled data customized for the particular task.

#### **2.3.2.1 Adding Prompts**

Creating a prompt involves modifying the input text through a two-step process. First, a template is applied—a string with placeholders for both the input text and a temporary answer, which will ultimately represent the desired output.

Placeholders can be embedded directly into the text or used as "prefix" prompts, where the text precedes the placeholder. They might also be represented by numerical codes or continuous vectors instead of natural language tokens. The quantity of placeholders can be modified according to the specific needs of various tasks.

#### **2.3.2.2 Search for Answers**

The objective is to determine the prediction with the highest score from the language model. This involves establishing a range of potential values for the intermediate generated response. In generative tasks, this range can encompass a wide variety of language options, whereas in classification tasks, it may be confined to a narrower set of terms. Next, the placeholder in the prompt is replaced with a candidate answer from this set, resulting in a completed prompt. If this completed prompt contains the correct answer, it is referred to as an "answered" prompt. The potential answers are then evaluated by assessing the likelihood of the completed prompts using pre-trained language models. This evaluation can be conducted through an argmax search, which identifies the highest-scoring output, or through sampling, which generates outputs based on the probability distribution of the language model.

#### 

#### **2.3.2.3 Mapping Answers**

In this context, the objective is to establish a correspondence between the response and the outcome. While in certain tasks like machine translation, the response can directly serve as the outcome, in other instances, several responses might align with similar outcome.

#### **2.3.2.4 Design Considerations for Prompting**

This segment delineates various designs employed in devising a methodology for prompting the tasks:

* Selection of Pre-trained Model: Several pre-trained LMs exist and can serve as foundational architecture for various NLP tasks.
* Crafting of Prompts: The selection of prompts holds considerable sway over the precision and character of the task executed by the model.
* Crafting of Responses: Depending on the task type, it might be imperative to devise a repertoire of potential values for the interim generated response and tailor the mapping function between responses and outputs accordingly.
* Expansion of the Framework: The foregoing sections offer a fundamental framework for prompting methodologies.
* Training Strategies Based on Prompts: For the language model, diverse strategies can be incorporated to for training.

### **2.3.3 Pre-trained Language Models**

Many extensive reviews have examined the effects of pre-trained language models (PLMs) within the "pre-train and fine-tune" framework (Qiu et al., 2020; Raffel et al., 2020; Doddapaneni et al., 2021). This section, however, will focus on analyzing different pre-trained language models specifically from the standpoint of prompting techniques.

#### **2.3.3.1 Training Objectives**

An approach known as denoising objective, involves introducing noise to input sentences and then attempting to reconstruct them to their original form. This method helps the model learn to handle corrupted or incomplete data by training it to recover the original content. Two prevalent denoising objectives include Corrupted Text Reconstruction (CTR) amd Full Text Reconstruction (FTR).

#### **2.3.3.2 Noising Techniques**

#### Various noising functions can be used, such as randomly masking entity names, substituting entities with generic tokens, or adding irrelevant words, each designed to challenge and refine the model's recognition and prediction capabilities, including:

#### Masking Noise: Randomly replacing certain words or entities with a special token (e.g., `[MASK]`), forcing the model to predict the original content based on the context.

#### Dropout Noise: Randomly dropping words or entities from the input during training to encourage the model to rely on the remaining context for predictions.

#### Perturbation Noise: Introducing small, controlled changes to the input entities or their contexts to help the model generalize better across different variations of the input.

#### Synonym Replacement: Replacing entities with their synonyms or similar words to test the model's ability to handle variations in entity representation.

#### Each type of noising function serves to challenge the model in different ways and can impact the effectiveness of the training process.

#### **2.3.3.3 Representation Directionality**

This consideration involves how the model processes and integrates information from different parts of the input text. Directionality impacts the model’s ability to capture contextual relationships and dependencies, which is essential for accurately interpreting and generating text. Broadly two prevalent approaches exist:

* Left-to-Right: Under this approach, the representation of each word is contingent upon the word itself and every preceding word in the sentence.
* Bidirectional: Here we rely on every other word in the sentence, encompassing both its left and right context.

Attention masking, a technique employed in the Transformer architecture, facilitates the calculation of representation directionality.

#### **2.3.3.4 Common Pre-training Approaches**

Some popular pre-training methods are mentioned below.

* Left-to-Right Language Models (LMs)
* Masked Language Models (MLMs):
* Prefix and Encoder-Decoder Language Models
* Encoding
* Decoding

By using prompting techniques, their versatility is further expanded, enabling unified modeling across different tasks by adjusting the model's outputs to fit specific prompts and contexts.

### **2.3.4 Prompt Engineering**

#### This process includes exploring and identifying optimal prompt templates tailored to the task. Prompt engineering involves crafting and refining prompts to elicit the most effective and accurate responses from language models. This process includes designing and optimizing the text used to interact with models, ensuring that the model's output aligns closely with the intended task or objective.

#### **2.3.4.1 Prompt Structure**

Prompt shape in prompt engineering refers to the design and structure of the input given to a language model to achieve desired responses. This includes elements such as the format of the prompt (e.g., questions, instructions, or examples), its length, and the clarity of the context provided (Petroni et al., 2019). Effective prompt shaping involves tailoring the input to ensure that the model understands the task and generates accurate outputs. For example, clear and specific prompts with well-defined examples can help guide the model's responses more effectively than vague or ambiguous ones. Additionally, the use of placeholders or special tokens within the prompt can help format the input in a way that aligns with the model's expectations and improves performance on a variety of tasks (Lester et al., 2021).

#### **2.3.4.2 Manual Template Design**

#### This approach was employed in developing the LAMA dataset for cloze prompts (Petroni et al., 2019). Similarly, manually designed prefix prompts have been used across various domains, including common sense reasoning, machine translation, and question answering.

#### **2.3.4.3 Automated Template Design**

These have been introduced to address the challenges of manual creation, including the time and expertise required and the difficulty experienced designers may face in identifying suitable prompts (Jiang et al., 2020a; b). These methods streamline the process by leveraging algorithms and data-driven approaches to generate effective prompts more efficiently.

Automated design methods can be classified into Discrete & Continuous Prompts and the functions can be classified as Static & Dynamic prompts

### **2.3.5 Answer Engineering**

#### It encompasses developing strategies and defining an answer space to navigate within that space to build an effective predictive model.

#### **2.3.5.1 Answer Format**

In NLP tasks, the response format influences the detail level and can vary in different ways. The response might be a simple classification label, a specific entity, a structured output like a list or table, or a more complex, free-form text. The choice of answer format is influenced by the task requirements and the nature of the information being extracted or generated. These may include:

#### Tokens

#### Multi-token spans

#### Sentences or documents

#### **2.3.5.2 Methods for Answer Format Design**

### Answer engineering involves defining the structure of answers, designing the answer space, and mapping answers to the output space. This process includes determining how answers should be formatted, organizing the possible types of responses, and ensuring that the generated answers align with the expected output for various tasks. Below are various methods for searching within this space:

### Exact Match Search: This method looks for answers that exactly match predefined entries in the answer space. It works well when answers are specific terms or phrases.

### Token-Based Search: This technique searches based on individual tokens or words. It is useful for tasks where answers are specific keywords or terms.

### Span-Based Search: This approach involves finding segments or spans of text. It is effective when the answer is a continuous segment of text rather than a single word. For example, in extractive summarization, the answer might be a phrase like "the capital of France," and span-based search will locate this entire segment within a larger text.

### Sequence Search: This method focuses on sequences of tokens or text spans. It is suited for tasks requiring recognition of coherent sequences or patterns. For example, in language generation tasks, where the goal is to create a coherent sequence of words, sequence search helps in constructing these sequences.

### Vector-Based Search: This technique uses vector embeddings to search the answer space. By converting text into numerical vectors, it measures semantic similarity between the query and potential answers. For example, if the query is "What is the tallest building?" and the answer space includes various building names, vector-based search helps identify the building most similar to "tallest."

### Probabilistic Search: This method uses probabilistic models to estimate the likelihood of different answers being correct or relevant. It ranks answers based on probability scores, prioritizing more likely answers. For instance, in multiple-choice question answering, probabilistic search can rank options based on their likelihood of being correct.

### Each method aids in customizing the answer space and search strategy to fit specific task requirements, enhancing the precision and effectiveness of response generation and evaluation.

### **2.3.6 Training Approaches for Prompting**

To proceed from the earlier discussions on prompts and answers, the next step is to develop and implement training strategies that effectively incorporate the chosen prompts with the models.

* Prompt-Model Integration: This involves incorporating the prompts into the model's training pipeline. Ensuring that the prompts are compatible with the model’s architecture and training objectives is essential for optimal performance.
* Fine-Tuning: This step focuses on further training pre-trained models using the prompts designed for specific tasks. Fine-tuning adjusts the model’s parameters to better align with the nuances of the task and the prompts.
* Training with Prompts: During the training phase, prompts are used as part of the input data for the model. This approach allows the model to learn how to process and respond to the prompts effectively.
* Evaluation of Prompting Strategies: Assessing the effectiveness of different prompting methods is crucial. This evaluation involves measuring the model’s performance based on accuracy, coherence, and relevance of the outputs produced in response to the prompts.
* Iteration and Optimization: The final step involves refining the prompts and training methods based on performance feedback. Iterative adjustments help in optimizing the model and prompts to achieve better results.

By integrating these strategies, you can enhance the model’s ability to perform specific tasks more effectively, leading to improved outcomes in natural language understanding and generation.

#### **2.3.6.1 Training Configuration**

#### There exist methodologies that involve fine-tuning the model alongside using prompts. These approaches can be categorized based on the quantity of training samples:

#### Full-data learning: This approach leverages extensive datasets to build robust models capable of capturing complex patterns and relationships.

#### Few-shot learning: Utilizes a limited number of examples. With a smaller dataset, models may struggle to perform as desired. Prompting methods can guide the model in such scenarios, helping to direct its behavior with the limited examples available (Liu et al., 2021b).

#### Prompting methods are especially useful in few-shot learning situations as they help improve model performance even with a small amount of training data.

#### **2.3.6.2 Methods for Parameter Updates**

### In downstream task learning using prompts, there are two main types of parameters to consider:

### Parameters from the pre-trained model.

### Parameters from the prompts.

### Selecting which parameters to update can significantly impact the model's performance. Careful choice of parameters to adjust during training influences how well the model learns from the data and adapts to various tasks, thereby affecting its overall effectiveness and accuracy.

### Promptless Fine-tuning:

### This method involves fine-tuning only the parameters of the pre-trained language model, without modifying the prompts. Examples include BERT and RoBERTa.

### It is effective for large datasets but may lead to overfitting or instability on smaller datasets, and could result in catastrophic forgetting.

### Tuning-free Prompting:

### This approach involves using prompts to generate answers without changing the parameters of the pre-trained model. It may include in-context learning, where input includes answered prompts. Examples include LAMA and GPT-3.

### This method is efficient and prevents catastrophic forgetting, making it suitable for zero-shot scenarios.

### Fixed-LM Prompt Tuning:

### In this approach, only the prompt parameters are adjusted based on supervision from downstream training samples, while the pre-trained language model parameters remain unchanged. Examples include Prefix-Tuning and WARP.

### It retains the language model's knowledge, making it appropriate for few-shot scenarios.

### Fixed-prompt LM Tuning:

### This approach allows the model to better adapt to specific tasks and prompts, improving its performance on tasks with limited training examples.

### It allows for more detailed task specification through prompt or answer engineering, leading to improved few-shot learning.

### Prompt+LM Tuning:

### This strategy involves updating the models by simultaneously refining the model and tailoring the prompts based on specific task data, this approach enhances the model’s ability to generate accurate and contextually relevant responses. Examples include PADA and P-Tuning.

### It is suitable for scenarios with sufficient data and involves training and storing all model parameters.

### **2.3.7 Prompt-Related Topics**

We explore how prompt-based techniques interact with various approaches, including supervised learning, transfer learning, and few-shot learning, to understand their combined impact on model performance and adaptability.

#### **2.3.7.1 Few-shot Learning**

#### A few-shot learning paradigm is designed to function effectively with a limited number of training samples, in contrast to supervised learning, which depends on a large volume of data. Several approaches facilitate few-shot learning, including meta-learning, where models are trained to quickly adapt to new tasks with minimal examples, and transfer learning, which leverages pre-trained models to apply learned knowledge to new, data-scarce tasks. Other methods involve prompt-based techniques that guide models in generating accurate responses even with sparse data by leveraging contextual cues and pre-trained knowledge.:

#### Embedding learning

#### Model-agnostic meta-learning

#### Memory-based learning

#### **2.3.7.2 QA-based Task Formulation**

Earlier research on question-answering (QA)-based prompting investigated the approach of reformulating a range of natural language processing (NLP) tasks as QA problems. This strategy aims to consolidate diverse tasks into a cohesive framework by leveraging text-based questions to define and direct the tasks. This technique is designed to take full advantage of the strengths of pre-trained language models. Despite these advancements, previous studies have not thoroughly exploited the full potential of these models, particularly in optimizing their application for QA-based formulations. There remains significant opportunity for further exploration and refinement in utilizing pre-trained language models to enhance QA-based prompting techniques and achieve more nuanced and effective task performance.

#### **2.3.7.3 Controlled Text Generation**

#### This can take various forms, such as constraints, prompts, or specific instructions, to ensure the generated content meets desired criteria or adheres to particular themes and styles. The various forms are:

#### Styling rules: Guidelines for maintaining a specific style in the generated text, as seen in the works of Sennrich et al. (2016) and Fan et al. (2018).

#### Domain tags: Labels indicating the context or field relevant to the text, used by Chu et al. (2017).

#### Length specifications: Instructions to generate text of a certain length, discussed by Kikuchi et al. (2016).

#### Keywords: Specific words that must be included in the output, highlighted by Saito et al. (2020).

#### Key phrases or sentences: Phrases that should be incorporated into the generated text, as demonstrated by Grangier and Auli (2017) and Liu and Nikolic (2021).

#### Relation triples: Relationships between entities that guide the text generation, used in the approach of Zhu et al. (2020).

#### Controlled text generation typically focuses on regulating aspects such as style or content while keeping the original task unchanged. This approach may not always require a pre-trained model, (Fan et al. 2018). In contrast, narrative generation using prompts leverages pre-trained models to define and specify the task itself, as discussed by Li and Liang (2021). Prompt-based methods generally utilize prompts at the dataset or task level, rather than the input-dependent prompts.

#### **2.3.7.4 Data Augmentation Techniques**

Techniques can include various transformations or alterations to the original data, such as paraphrasing, adding noise, or applying other modifications to create new training examples.

Research by Le Scao and Rush (2021) shows that incorporating prompts into a dataset can have a similar effect on model accuracy as adding a substantial number of new data points. This finding underscores the potential of prompts to enhance model performance in a manner comparable to traditional data augmentation methods. By providing additional context or guiding the model's learning process through prompts, one can achieve performance improvements similar to those obtained through more extensive data augmentation strategies.

### **2.3.8 Challenges in Prompting**

Prompting faces several challenges, including the need for precise formulation to achieve reliable results, as slight changes in phrasing can significantly affect the output. Additionally, ensuring that prompts are effective across a range of tasks and contexts requires extensive experimentation. Models might also misinterpret ambiguous instructions, leading to inconsistent or suboptimal responses. There are several challenges that need to be addressed:

#### **2.3.8.1 Designing Effective Prompts**

Most studies on prompt-based learning have primarily focused on tasks like text classification and generation, leaving information extraction and text analysis underexplored. This disparity may be due to the added complexity of crafting effective prompts for these tasks. Future research could explore reframing these challenges as classification or generation problems or employing answer engineering techniques to improve the presentation of structured outputs.

The success of prompt-based models hinges on both the prompt design and the generated answer, creating the challenge of optimizing both elements simultaneously. Some studies have concentrated on choosing answers before crafting templates (Gao et al., 2020c; Shin et al., 2020), while others, like Hambardzumyan et al. (2021), have shown that it is feasible to learn both the template and answer together.

#### **2.3.8.2 Engineering Answer Formats**

We are facing two main challenges in our research work:

* Managing a large number of classes can complicate the process of defining an appropriate answer space. The diverse range of possible answers makes it challenging to establish a comprehensive and effective answer set.
* The use of multi-token answers can be problematic when decoding multiple tokens with language models. Jiang et al. (2020b) proposed solutions to address these difficulties, aiming to improve how models handle and generate multi-token responses.

For text generation tasks, even when responses are semantically similar, variations in syntactic structure can occur. Research has predominantly focused on using a single reference answer, but current studies are investigating ways to guide the learning process more effectively when multiple references are available. This involves developing methods to manage and utilize multiple textual references to enhance the quality and consistency of generated text.

#### **2.3.8.3 Choosing Tuning Strategies**

Various strategies are available for fine-tuning prompt parameters, language model parameters, or both simultaneously. Each approach offers distinct benefits and challenges. For instance, adjusting prompt parameters allows for targeted customization of how the model responds to specific inputs, which can enhance relevance and accuracy. Conversely, tweaking language model parameters can optimize the model’s overall performance, potentially improving its ability to generate coherent and contextually appropriate outputs. Combining adjustments to both prompts and model parameters can provide a more comprehensive solution but may also introduce complexity in balancing and managing these changes. Each method needs to be evaluated based on the specific requirements of the task and the desired outcomes to determine the most effective approach.

#### **2.3.8.4 Learning with Multiple Prompts**

Prompt composition and decomposition are methods for managing complex tasks by breaking them down into simpler sub-prompts or combining them into more comprehensive ones. The optimal choice between decomposition and composition is not always clear.

Prompt augmentation refers to enhancing the input with additional samples to improve model performance. However, these techniques are limited by the length of the input, necessitating further research into how to select and organize informative samples effectively.

Prompt sharing involves applying learned prompts across different domains, tasks, or languages. Despite its potential, research in this area is still scarce. The methods for creating task-specific prompts or managing their interactions are not yet fully understood, highlighting a need for more exploration in this domain.

#### **2.3.8.5 Selecting Pre-trained Models**

In the field of prompt-based learning, choosing the most suitable pre-trained language model from a variety of options is challenging. While there are broad guidelines and theoretical frameworks for choosing models based on particular NLP tasks, there is a notable absence of detailed, systematic comparisons that examine how various pre-trained language models perform when applied with prompts. This gap underscores the need for comprehensive studies to evaluate and compare the effectiveness of various pre-trained models in prompt-based learning contexts.

#### **2.3.8.6 Prompt Transferability**

Understanding how specific prompts perform with different pre-trained models is essential for improving prompt transferability. Research by Perez et al. (2021) highlights the following insights:

* In scenarios involving fine-tuned few-shot learning with access to a substantial validation dataset for prompt selection, prompts typically yield favorable outcomes across models of similar sizes. This suggests that with adequate fine-tuning and validation, prompts can be effectively adapted to different models within a similar size range.
* In true few-shot learning scenarios, where training samples are scarce, prompts demonstrate limited effectiveness in generalizing to models of comparable sizes. This indicates that with limited training data, prompts may not transfer well between models.
* Prompts designed for one model may not work effectively with models of different sizes, highlighting the need for model-specific prompt design.

#### **2.3.8.7 Combining Different Paradigms**

There is significant uncertainty about whether the pre-training methods that have proven effective for the pre-train and fine-tune strategy are equally applicable to prompt-based learning. As prompt-based learning becomes increasingly popular, it is crucial to investigate whether novel pre-training methods could improve prompt-based strategies or if existing methods can be adapted for this purpose.

#### **2.3.8.8** **Calibrating Prompting Methods**

#### Pre-trained language models often struggle with calibration, especially when generating answers. Zhao et al. (2021) highlighted three key biases that can affect these models:

#### Recency Bias: The model may disproportionately favor more recent information or tokens, influencing the generated probabilities and responses towards these recent elements.

## Majority Label Bias: The model might show a preference for more common labels or tokens encountered during training, even if these are not the most suitable for the current context.

## Common Token Bias: The model may generate responses that lean towards frequently occurring tokens or phrases, which can affect the diversity and accuracy of the outputs.

## Understanding and mitigating these biases is crucial for enhancing the performance and reliability of pre-trained language models.

## **2.4 Summary**

Extensive research has been carried out on automatic story generation, often using older models such as GPT-2. However, the capabilities of newer models like GPT-3 and its successors in this field have not been thoroughly explored.

Prompt-learning, an emerging technique in natural language processing (NLP), has garnered significant attention for its applications across various tasks, particularly in text generation. Despite this growing interest, there is a distinct lack of research dedicated to leveraging prompt-learning for narrative story generation. This gap highlights an opportunity for further exploration into how prompt-learning can be effectively utilized to enhance the creation of narratives, potentially leading to novel approaches and improvements in storytelling through automated systems.

This study seeks to address these gaps by applying prompt-learning techniques to the task of story generation. It seeks to provide benchmark scores using advanced language models from the latest generation, thus advancing the field and offering fresh insights into the capabilities of contemporary pre-trained models for narrative creation.

# **CHAPTER 3**

# **RESEARCH METHODOLOGY**

## **3.1 Overview**

This study aims to assess the effectiveness of Pre-trained Language Models (PLMs) in two key areas: story generation and prompt-learning. Specifically, it evaluates how well these models can generate coherent and engaging narratives, as well as their ability to leverage prompting techniques to improve their performance. By examining these aspects, the study seeks to gain insights into the strengths and limitations of PLMs in creating and refining stories, and how different prompting strategies can influence their output. The following sections will explore the fundamental concepts and algorithms critical to this research.

## **3.2 Algorithms and Techniques**

### **3.2.1 Pre-Trained Language Models**

Large Language Models (LLMs), also known as Pre-trained Language Models (PLMs), are designed with the Causal Language Modeling (CLM) objective, which focuses on predicting the next word in a sequence. This foundational capability makes LLMs crucial for transfer learning applications. Recent progress in Large Language Models (LLMs) and the development of frameworks like Prompt-Learning have greatly improved their performance in Few-Shot and Zero-Shot Learning situations. These advancements have enabled LLMs to better understand and generate relevant responses with minimal examples or even without prior specific training on the task at hand. These developments enable LLMs to perform complex tasks with minimal task-specific training, showcasing their versatility and power in natural language processing.

#### **3.2.1.1 OPT**

Released by Facebook/Meta in 2022, they share a similar architecture with GPT-3 and focus on Causal Language Modeling (CLM) as their core task. These models are assessed using prompts and experimental frameworks similar to those applied to GPT-3. Trained on a vast dataset of approximately 800GB, which includes sources such as Book Corpus, CC-Stories, The Pile, Pushshift.io Reddit dataset, and CCNewsV2, the OPT models offer a broad range of parameter sizes.

#### **3.2.1.2 OPT-IML**

Facebook's OPT models were further refined through Instruction Tuning using a method known as Instruction Meta-Learning (IML), as described by Iyer et al. (2022). The OPT-IML models are trained on an extensive benchmark called OPT-IML Bench, which includes 2,000 NLP tasks. It combines data from eight established benchmarks, including Super-Natural Instructions, FLAN, and Prompt Source, among others. For instance, the OPT-30B model underwent fine-tuning using 64 40GB A100 GPUs.

#### **3.2.1.3 LaMDA & PaLM**

Google has developed several state-of-the-art Large Language Models, including LaMDA and PaLM. LaMDA, detailed by Thoppilan et al. in 2022, is specifically designed to excel in dialogues and open-ended conversations. It focuses on enhancing the model’s ability to maintain natural and engaging interactions across a wide array of conversational topics.

PaLM, introduced by Chowdhery et al. in 2022, is a dense decoder-only Transformer model with an extensive 540 billion parameters, positioning it among the largest language models available. Despite its advanced capabilities, Google has not made PaLM or LaMDA available as open-source resources or commercial APIs. For further information on related developments and advancements, see Section 2.2.3.

#### **3.2.1.4 GPT-2**

This transformer-based model was trained on an extensive corpus of English text and is particularly proficient in text generation tasks using prompts. Developed by OpenAI, GPT-2 was trained on the WebText dataset, which includes content from 45 million outbound links shared on Reddit. The dataset was pre-filtered to remove sexually explicit or offensive material.

#### **3.2.1.5 GPT-3**

GPT-3 represents a significant advancement over GPT-2, featuring a remarkable 175 billion parameters. This model was trained on a diverse and extensive dataset totaling approximately 45 terabytes of text data from various internet sources:

* Common Crawl: A dataset that includes petabytes of data collected from web crawling over an 8-year period, as provided by Commoncrawl.org (2023).
* WebText2: An updated and expanded version of the Web Text dataset used for GPT-2, incorporating additional data gathered after GPT-2's release.
* Books1 and Books2: Datasets comprising text extracted from books available online.
* Wikipedia: A collection of English-language Wikipedia pages, which provides a broad range of textual content.

These diverse sources contribute to GPT-3's extensive knowledge and its ability to generate high-quality text based on prompts.

Different GPT model sizes:

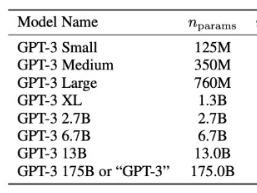


Figure 3.2.1.1: GPT-3 Model Sizes

#### **3.2.1.6 GPT-3.5**

Causal Language Models, such as GPT-3, generate text by predicting the next word based only on the preceding words, which can sometimes lead to outputs that do not fully align with user instructions. In Prompt-Learning, especially in Zero-Shot scenarios, the effectiveness of the results depends significantly on the quality of the initial prompts provided. To address this issue, OpenAI introduced the InstructGPT model, which fine-tunes GPT-3 using Reinforcement Learning from Human Feedback (RLHF) to enhance its ability to follow instructions and reduce the occurrence of factually incorrect outputs.

#### **3.2.1.7 BLOOM**

Training of LLMs is indeed a resource-intensive process, requiring extensive datasets and significant computational resources. This poses a barrier to many researchers and organizations, limiting access to these advanced models and potentially impeding progress in related research areas. Moreover, the predominance of English and a few other high-resource languages in available text data means that LLMs tend to perform better in these languages compared to low-resource languages. Fine-tuning these models for low-resource languages is particularly challenging due to the associated costs. To tackle these challenges, the BigScience initiative was established as a large-scale collaborative project led by HuggingFace, GENCI, and IDRIS. This effort united over 1,000 researchers from diverse domains.

#### **3.2.1.8 BLOOMZ**

Created by Muennighoff et al. in 2022, BLOOMZ models are specialized variants of the BLOOM language models that have been fine-tuned for instruction following. These models are designed to comprehend and act on human instructions across a broad spectrum of languages, leveraging zero-shot learning capabilities without requiring specific prior training.

The BLOOMZ models were fine-tuned using the xP3 dataset, which stands for Crosslingual Public Pool of Prompts. This dataset encompasses prompts and data for 46 languages and 16 distinct NLP tasks, facilitating extensive cross-lingual training and evaluation.

### **3.2.2 n-Shot Learning**

While Transfer Learning helps alleviate the need for large amounts of labeled data, high accuracy in specific domains often still requires several hundred labeled examples per class, which can be costly to produce. In the n-Shot learning framework, which encompasses Few-Shot, One-Shot, and Zero-Shot learning, the objective is to achieve accurate predictions with minimal samples, without adjusting the base model for specific tasks or datasets. During the inference phase, the model parameters remain static, and predictions are guided by reference samples.

## **3.3 Methodology**

### **3.3.1 Comprehensive Pipeline**

Complete pipeline for the methodology is shown in Figure 3.3.1.1.

Diagram

Description automatically generated

Figure 3.3.1.1: Comprehensive Pipeline

### Complete pipeline for story generation involves several steps:

### Data Corpus Selection: The process begins with choosing a data corpus from sources like ROCStories or WritingPrompts. This corpus provides the base material for story generation.

### Story Extraction: Stories are extracted from the selected corpus. This step collects the raw stories that will be used in the following stages.

### Outline Extraction: Outlines are derived from each story. These outlines act as summaries or blueprints of the story’s main components and themes.

### Prompt Creation: The extracted outlines are used to generate prompts. These prompts guide the language models by providing specific instructions or cues for creating new stories.

### Story Generation: Using the prompts, the language models generate new stories. The prompts help the models follow the structure and elements outlined in the previous step.

### Automated Evaluation: The generated stories are assessed using automated metrics. These metrics evaluate various aspects such as coherence, relevance, and fluency of the stories.

### This pipeline allows for the systematic creation and evaluation of stories through the use of prompts and language models.

### **3.3.2 Description of Data**

This work utilizes two datasets most commonly used for story generation:

* ROCStories
* WritingPrompts

**3.3.3 Data Preparation Process**

#### **3.3.3.1 ROCStories Dataset**

The ROCStories dataset consists of a collection of 5-sentence stories designed to provide simple, coherent narratives for evaluating story generation models. Each story in the dataset is structured with a clear, logical progression of events, making it a valuable resource for testing how well models can generate or understand short narratives. The dataset is primarily used to evaluate how effectively models can extend a given story outline or generate a complete story based on minimal input. ROCStories is commonly utilized in research to assess the quality of automated story generation systems and their ability to maintain narrative coherence. It is well-regarded for its consistency and ease of use in benchmarking various natural language processing tasks related to storytelling.

Diagram

Description automatically generated

Figure 3.3.3.1: ROCStories Outline Generation

#### **3.3.3.2 WritingPrompts Dataset**

The WritingPrompts dataset is a collection of creative writing prompts designed to inspire and generate diverse narratives. It contains pairs of prompts and corresponding stories, providing a wide range of starting points and thematic elements for narrative development. Each entry in the dataset includes a prompt, which is typically a brief, open-ended statement or question, and a full-length story written in response. This dataset is valuable for training and evaluating models on tasks related to creative writing and story generation. It is commonly used to assess how well language models can generate coherent and engaging text based on various prompts.

Diagram

Description automatically generated

Figure 3.3.3.2: WritingPrompts Outline Generation

### **3.3.4 Implementation Details**

#### **3.3.4.1 Creating Prompts from Outlines**

In this phase, the previously created story outlines are transformed into prompts for Prompt-Learning. For Zero-Shot Prompting, an introductory instruction is appended to the start of the outline. For instance, the instruction could be: "Create a story based on the following outline:"".

Diagram

Description automatically generated

Figure 3.3.4.1: Zero-Shot Prompt

In Few-Shot Prompting, the prompt includes several examples of outline-to-story pairs, accompanied by an initial instruction. The final example in this series contains just the outline, and the model is tasked with generating a story based on this concluding outline.

Diagram

Description automatically generated

Figure 3.3.4.2: Few-Shot Prompt

#### **3.3.4.2 Generating Stories from Prompts**

The prompt generated in the earlier step is input into the model for processing. The model then produces a story paragraph based on the provided outline.

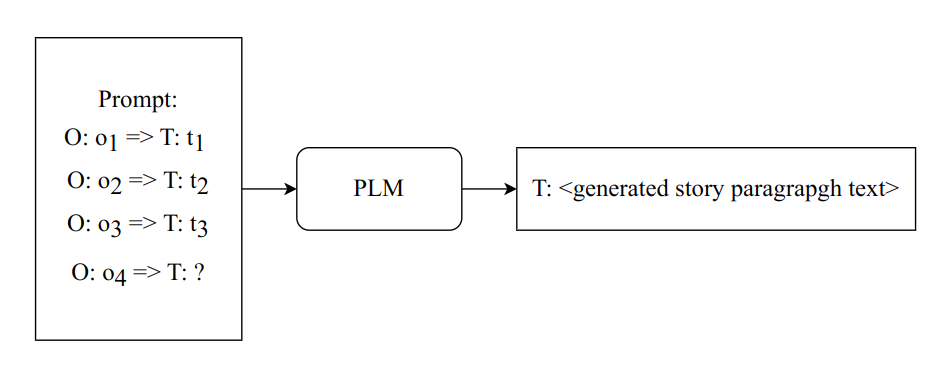


Figure 3.3.4.3: Prompting for Story Generation

### **3.3.5 Evaluation**

## Output is assessed using several automatic metrics such as:

## Perplexity (PPL): This quantifies the exponentiated average negative log-likelihood of a sequence, indicating how likely the model believes the sequence to be. Lower perplexity indicates better performance. However, this metric is influenced by the training data and may not be directly comparable across different models or datasets.

## BLEU (Bilingual Evaluation Understudy): BLEU evaluates the overlap of n-grams between the generated text and reference texts. It is commonly used in machine translation to gauge how similar the generated text is to human-written references. Although a higher BLEU score suggests better similarity, BLEU has limitations such as focusing on token overlap rather than semantic meaning and preferring shorter, more exact matches.

## ROUGE (Recall-Oriented Understudy for Gisting Evaluation): ROUGE measures the quality of summaries or translations by comparing generated outputs to reference texts. It is used to evaluate the coverage and relevance of the generated content relative to human references.

## BERTScore: This metric utilizes contextual embeddings from BERT to measure cosine similarity between words in generated sentences and reference sentences. It provides precision, recall, and F1 scores, which closely match human judgments for both individual sentences and overall system evaluations.

## **3.4 Summary**

# This chapter detailed the algorithms and techniques applied in the thesis, explaining their methodologies thoroughly. It also offered an extensive overview of the implementation process, covering the hardware and software tools utilized as described in Chapter 4.

# **CHAPTER 4**

# **IMPLEMENTATION**

## **4.1 Overview**

## This chapter provides a detailed explanation of the code-level implementation aspects. We have covered the following sections in this chapter:

## Data Preparation: This section delves into the chosen dataset and explores the various preprocessing procedures executed on the dataset.

## Story Generation: This section provides a comprehensive overview of all the tests performed in this thesis.

## **4.2 Data Preparation**

Outlined in section 4.3.3, challenges arose during the utilization of the WritingPrompts dataset. Consequently, the decision was made to opt for the ROCStories dataset for this thesis. The DAG conducted for preparing the data is show below.

Diagram

Description automatically generated

Figure 4.2.1: Data Preparation Steps

The steps conducted are elaborated as follows:

1. From the website, download the files
2. Selection of Validation Set – 2018: We selected the Winter 2018 dataset from the provided choices, as it is the latest and recommended by the authors.
3. Sampling the First 100 Records: To manage costs associated with GPU usage and OpenAI API calls, restrict the story generation process to the initial 100 stories from the Winter 2018 validation set.
4. Selection of the 5th Sentence: The dataset, created for the Story Cloze test, includes two potential choices for the 5th sentence—RandomFifthSentenceQuiz1 and RandomFifthSentenceQuiz2.
5. Creating the Outline: Sentences are merged using newline (\n) characters to form the story outline. This outline is then added to a new column titled "outline." A sample of the generated outline is shown in Figure I.

|  |
| --- |
| Tom discovered an old disposable camera buried at the bottom of his cluttered drawer.  He started taking pictures of everything in sight.  The counter showed only one photo left.  Tom began to realize the significance of the moment.  Tom paused to consider what he would photograph next. |

Figure 4.2.2: Sample Outline

1. Creating the Prompt: Refine the generated outline by incorporating detailed instructions and relevant tags to formulate a prompt that can be effectively used as input for the model. A completed prompt is shown below.

|  |
| --- |
| Generate a multi-paragraph story using the following outline:  Tom discovered an old disposable camera buried at the bottom of his cluttered drawer.  He started taking pictures of everything in sight.  The counter showed only one photo left.  Tom began to realize the significance of the moment.  Tom paused to consider what he would photograph next. |

Figure 4.2.3: Sample Prompt

## **4.3 Story Generation Process**

### **4.3.1 Procedures for Generating Stories**

The steps performed are shown below:

Diagram

Description automatically generated

Figure 4.3.1.1: Story Generation Steps

Complete procedure is described as follows:

1. Obtaining the Prompt: Retrieve the prompt that was generated during the data preparation stage.
2. Using the API of OpenAI: Implement two approaches for generating stories:
3. Story Generation:
   1. API of OpenAI: Utilize the Python library from OpenAI to interact with different pre-trained GPT-3 models. This service requires payment, but OpenAI offers initial free credits for new users.
   2. Huggingface: Use Huggingface's Python Library to access various GPT-3-like models available on the Huggingface hub.
4. Selecting the Best from the Generated Stories: Select the longest story as the most favorable one, as length often correlates with higher quality. Shorter stories might only replicate the outline, suggesting they may be of lower quality.
5. Repeating for Models of different types: Perform all the above steps for various models

### **4.3.2 Zero-Shot Story Generation Techniques**

The procedures described in Section 4.3.1 were performed using a zero-shot approach, applying the zero-shot story generation method to all models listed in Step 5 of that section. Detailed analysis of the story evaluation results for these models is presented in Chapter 5.

Moreover, zero-shot generation was also conducted on the WritingPrompts dataset, where story outlines were extracted from existing narratives. Despite employing both zero-shot and n-shot techniques, the quality of the generated outlines was notably poor, leading to a decline in the quality of the resulting stories. Due to the substantial difference in generation quality compared to the ROCStories dataset, these results were excluded from the evaluation in Chapter 5.

### **4.3.3 One-Shot and Few-Shot Story Generation Techniques**

Creating stories through One-Shot and Few-Shot techniques poses several difficulties, detailed as follows:

* The dataset includes 5-sentence stories intended as outlines for developing longer narratives, but lacks corresponding target stories that could serve as references for One-Shot and Few-Shot generation scenarios.
* WritingPrompts dataset was also used for One-Shot and Few-Shot experiments. The longer stories in this dataset allowed for the generation of outlines and the use of actual stories as targets for generation in these scenarios. Nevertheless, significant issues arose during story generation. The inclusion of these stories in the prompts resulted in very large prompt sizes. The issue was especially severe in Few-Shot generation, where the prompts included multiple stories, intensifying the memory constraints and leading to CUDA out-of-memory errors.

## **4.4 Summary**

This chapter delved into the critical data preprocessing techniques necessary for effective prompt learning, focusing on how these methods facilitate story generation in a Zero-shot context. It outlined the various steps involved in preparing the data, emphasizing how these steps ensure the model can generate coherent and contextually relevant narratives without prior specific examples.

The chapter also examined the challenges associated with One-Shot and Few-Shot generation approaches. It highlighted the limitations and complications encountered with these methods, which led to their exclusion from the evaluation discussed in Chapter 5. By addressing these issues, the chapter provided a comprehensive understanding of why Zero-shot learning was favored in this context and how the preprocessing steps contributed to its success.

# **CHAPTER 5**

# **RESULTS & EVALUATION**

## **5.1 Overview**

This chapter presents the findings from the experiments detailed in Chapter 4, encompassing both quantitative and qualitative assessments:

* Quantitative Evaluation
* Qualitative Evaluation

## **5.2 Quantitative Analysis**

Table A displays the performance scores across various metrics for stories generated by different models.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Babbage | Curie | Davinci | Bloomz-1b7 | Ada | Gpt2 |
| Bertscore\_mean\_recall | 0.91 | 0.91 | 0.91 | 0.85 | 0.89 | 0.78 |
| Bertscore\_mean\_precision | 0.85 | 0.83 | 0.76 | 0.82 | 0.81 | 0.71 |
| RougeLsum | 0.51 | 0.44 | 0.28 | 0.44 | 0.46 | 0.15 |
| Rouge2 | 0.36 | 0.28 | 0.15 | 0.29 | 0.32 | 0.04 |
| Perplexity | 27.75 | 25.83 | 32.67 | 91.77 | 70.77 | 18.89 |
| Rouge1 | 0.55 | 0.49 | 0.3 | 0.46 | 0.47 | 0.16 |
| Bleu | 0.22 | 0.17 | 0.08 | 0.23 | 0.2 | 0.02 |
| Bertscore\_mean\_f1 | 0.88 | 0.87 | 0.83 | 0.83 | 0.85 | 0.74 |
| RougeL | 0.45 | 0.39 | 0.24 | 0.38 | 0.39 | 0.11 |

Table 5.2.1: Evaluation Scores

Here are several important aspects concerning the calculation of these scores:

* Metrics other than perplexity were determined by comparing the generated stories against the outlines used for their creation.
* Perplexity was measured using the distilgpt2 model as a reference.
* The evaluation was conducted with the "evaluate" Python library.
* Scores were rounded to two significant figures to ensure clarity and consistency.
* For BERTScore, individual precision, recall, and F1 scores were averaged to obtain the overall mean values.

Figure 5.2.1 presents these findings, highlighting a noticeable trend as we progress from the GPT-2 model to the Davinci model. The x-axis of the figure displays the models in ascending order of complexity and size.

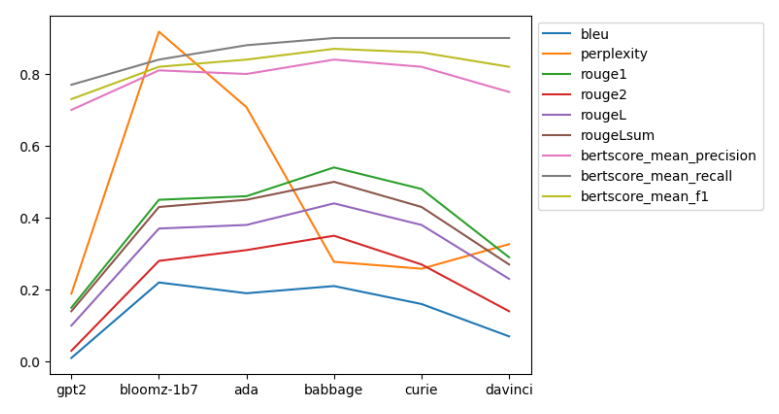


Figure 5.2.1: Evaluation Scores Plot

Above graph shows distinct trends across various metrics, with Perplexity displaying an inverse relationship compared to the others. Here's a comprehensive analysis:

* GPT-2 Model: Exhibits weaker performance in zero-shot generation because of its smaller size and outdated architecture compared to newer models. As a result, GPT-2 records the lowest scores across all metrics. Bloomz-1B7 Model: Although relatively small with 1 billion parameters, it shows markedly better performance than GPT2. Stories generated by Bloomz-1B7 align closely with outlines, leading to higher scores in BLEU, ROUGE, and BERTScore.
* Ada and Babbage Models: These smaller OpenAI models exhibit similar performance trends, with a steady improvement up to the Babbage model. The generated stories closely match the outlines in terms of size and content, resulting in consistent score increases.
* Curie and Davinci Models: As larger models, the stories produced are more elaborate and diverse. Despite maintaining adherence to the outlines, these models exhibit greater creativity, which leads to a decrease in scores compared to smaller models.
* Perplexity: Calculated using the distilgpt2 model, shows a trend where smaller models tend to have similar word probability distributions. As model size increases, and with a more extensive vocabulary and creative output, perplexity decreases.

## **5.3 Qualitative Analysis**

Here are two outline examples accompanied by the stories generated by different models based on those outlines:

### **5.3.1 Example of Generated Story – 1**

|  |
| --- |
| Carlos wants to join a sport while he’s in college.  Since he was an excellent swimmer, he decides to audition for the swim team.  Carlos easily makes the team.  At the first swim meet, Carlos secures second place!  Carlos earns a silver medal. |

Figure 5.3.1.1: Outline – 1

|  |
| --- |
| This tale concludes at the first round of the French Open.  The narrative begins with a girl walking into a bar.  It finishes with Carlos's head soaring through the air while he’s at the bar.  In the fourth round, Carlos earns a bronze medal. In the fifth round, Carlos claims another bronze medal.  In the sixth round, Carlos secures yet another bronze medal.  The story wraps up with a recount from the team.  The story opens with a group of girls heading to the beach in the park of Saint Pompon.  In the sixth round, the girls are still at the beach. By the seventh round, they remain at the beach.  One of them exclaims, “This is fantastic, I’m so happy.”  Soon after, Carlos is seen in a pool, floating underwater.  There is a notably awkward moment in the story. It concludes with a girl saying, “A very proud girl” when she remarks, “I’m so proud of you.”  Here are a few interesting notes:  The story begins in Saint Pompon.  It ends with a brief narrative.  And here is a brief narrative: |

Figure 5.3.1.2: Generated Story: GPT2 – 1

|  |
| --- |
| At the initial meet, Carlos swims exceptionally well. He performs well in running events too. He avoids taking any risks and wins with ease. The team is quite astonished by his performance at the first swim meet. Carlos avoids any risks and secures an easy victory. The team is thoroughly impressed by his debut performance. |

Figure 5.3.1.3: Generated Story: Bloomz-1B7 – 1

|  |
| --- |
| Carlos is in college and is a swimmer. He wants to join a sport, so he auditions for the team. He easily secures a spot on the team. At the first swim meet, Carlos finishes in second place! This achievement makes him extremely proud, and he is thrilled to finally be competing in a competition! |

Figure 5.3.1.4: Generated Story: Ada – 1

|  |
| --- |
| Carlos was thrilled to try out for the swim team during his first year of college. He quickly made the team and excelled, earning second place in his division. At the next meet, Carlos won a silver medal. His success continued as he accumulated more medals and trophies, eventually setting a world record in a major event. With this new achievement, Carlos became a top swimmer and earned a scholarship to college. |

Figure 5.3.1.5: Generated Story: Babbage – 1

|  |
| --- |
| Carlos had loved swimming ever since he became proficient in it. He had always wanted to join a swim team, so when the chance arose while he was in college, he seized it. He easily made the team and at his first swim meet, he achieved second place! Carlos was overjoyed and excited about his success. However, his happiness was further amplified when he discovered that he had also won a silver medal. This accomplishment made him even prouder and boosted his confidence for future swim meets. |

Figure 5.3.1.6: Generated Story: Curie – 1

|  |
| --- |
| Carlos had been contemplating playing a sport in college. Although he had participated in various sports during high school, swimming had always been his favorite. With this in mind, he decided to try out for the swim team at his college.  Upon arriving for the tryouts, Carlos saw several swimmers already competing and realized it would be tough competition. However, Carlos was an accomplished swimmer and performed well during the tryouts. After being assessed by the coach and demonstrating his skills in the pool, Carlos easily made the team.  At their first swim meet of the season, everyone was eager to see how they would perform as a team. With strong competitors in the mix, the team knew they had their work cut out for them! During one of his events, Carlos impressed everyone by finishing in second place! The team cheered as he received a silver medal and brought pride to them all.  Carlos was proud of himself for making it onto a top college swim team and performing so well at their initial meet of the year—an achievement few can claim! He now looked forward to pushing himself further with this new opportunity. Carlos had been contemplating playing a sport in college. Although he had participated in various sports during high school, swimming had always been his favorite. With this in mind, he decided to try out for the swim team at his college.  Upon arriving for the tryouts, Carlos saw several swimmers already competing and realized it would be tough competition. However, Carlos was an accomplished swimmer and performed well during the tryouts. After being assessed by the coach and demonstrating his skills in the pool, Carlos easily made the team.  At their first swim meet of the season, everyone was eager to see how they would perform as a team. With strong competitors in the mix, the team knew they had their work cut out for them! During one of his events, Carlos impressed everyone by finishing in second place! The team cheered as he received a silver medal and brought pride to them all.  Carlos was proud of himself for making it onto a top college swim team and performing so well at their initial meet of the year—an achievement few can claim! He now looked forward to pushing himself further with this new opportunity. |

Figure 5.3.1.7: Generated Story: Davinci – 1

### Figure 5.3.1.1 presents Outline-1, a 5-sentence plot from the ROCStories dataset.

### Figure 5.3.1.2 shows the story generated by the GPT2 model. This story demonstrates poor coherence and narrative flow, resembling incoherent text despite incorporating some elements from the outline. This indicates that adjustments to the model's hyperparameters may be necessary to improve story generation. Alternatively, fine-tuning GPT2 with outline-story pairs, as suggested by Fang et al. (2021), could potentially enhance its performance in generating conditioned outputs.

### In Figure 5.3.1.3, the story generated by the Bloomz-1B7 model is displayed. Although the story follows the outline closely, it primarily rephrases the outline with minimal addition of new content, resulting in repetition and limited originality. Utilizing a larger Bloomz model version might help overcome these issues.

### Figures 5.3.1.4, 5.3.1.5, and 5.3.1.6 show the outputs from the Ada, Babbage, and Curie models, respectively. There is a noticeable improvement in story quality from Ada to Curie, with the narratives maintaining coherence and aligning well with the outline. Particularly, the Curie model enriches the story with additional descriptive details.

### In contrast, Figure 5.3.1.7 presents the story generated by the Davinci model, which shows a significant improvement over previous model. The narrative demonstrates a human-like quality, structured into multiple paragraphs that develop distinct ideas while remaining interconnected. Additionally, the Davinci model introduces novel and imaginative concepts, enriching the overall story composition.

### **5.3.2 Example of Generated Story – 2**

|  |
| --- |
| Alex had just begun fifth grade.  He attempted to befriend the other boys, but struggled to connect with them.  He then chose to reach out to the girls instead.  The girls warmly accepted him into their social circles.  Meanwhile, Alex faced teasing from the other boys at school. |

Figure 5.3.2.1: Outline – 2

|  |
| --- |
| The other boys tried to take advantage of Alex, but he ignored their attempts. One of these boys started bullying the girl who had initially targeted him.  Then there was a boy who was involved in this behavior, with some help from Alex's father.  As Alex began making more friends, this boy started bullying him as well, though he never physically attacked Alex’s friend. The girl who had begun the bullying continued trying to get Alex to stop, while the boy fought to get her to cease her bullying, but he failed.  Despite his efforts, the bullying continued to escalate. Eventually, Alex managed to convince the girl to stop, but the boy began attacking her. This led to a confrontation between Alex and the girl, and when Alex fought back, he ended up hitting her.  Alex, his father, and the girls in the group eventually became friends, bonding over their mutual support and understanding.  However, Alex started losing friends as his behavior deteriorated. He was alienating people because of his actions. He began hitting others, but he didn’t hit the girl he was supposed to be defending. His aggression seemed misplaced and was affecting those around him.  Despite this, when the girl he cared about was involved, Alex’s anger became more focused. He continued to punch and fight, but it was clear that his actions were causing harm to everyone involved.  The boy who had been bullying Alex eventually became friends with the girl, while Alex's own friends, including his parents, began to distance themselves. The cycle of violence and aggression only seemed to spiral further, affecting all their relationships. |

Figure 5.3.2.2: Generated Story: GPT2 – 2

|  |
| --- |
| Alex went home in tears. The girls and the other boys started laughing at him. He decided to make friends with the girls instead. The girls welcomed him into their social circles with warmth and enthusiasm. |

Figure 5.3.2.3: Generated Story: Bloomz-1B7 – 2

|  |
| --- |
| Alex had done his best to befriend the other boys at school, but he just couldn’t connect with them.  The other boys picked on him, adding to the difficulties in his life.  However, when he visited the girls' social groups, he noticed they were thriving. Alex decided to join a social group for girls, and he was warmly welcomed into their circle.  Alex enjoyed spending time with his new friends and took pleasure in being one of the few boys in the group. |

Figure 5.3.2.4: Generated Story: Ada – 2

|  |
| --- |
| Alex tried to befriend the other boys at school, but he wasn’t successful. He chose to make friends with the girls instead, and they welcomed him into their social groups with enthusiasm. However, the other boys at school began to pick on Alex and make fun of him. Alex felt isolated and abandoned by his new friends. |

Figure 5.3.2.5: Generated Story: Babbage – 2

|  |
| --- |
| Alex had always been a shy individual. He worried that he wouldn't be able to make friends or fit in with the other boys in his class. However, after some trial and error, Alex realized that connecting with the girls was the best approach for him. They welcomed him warmly into their groups and didn’t treat him like an outsider. Alex learned that having a balance of both male and female friends is crucial for feeling comfortable in any social setting. As long as Alex maintains this balance, he will have a positive social experience no matter where life takes him. |

Figure 5.3.2.6: Generated Story: Curie – 2

|  |
| --- |
| Alex had just started fifth grade. As the new kid, he wasn’t sure how to fit in. He tried to befriend the other boys, but they were not very receptive. No matter how hard Alex tried, the boys always treated him with cold indifference, leaving him feeling lonely and isolated.  Desperate for companionship, Alex decided to try a different approach and make friends with the girls instead. To his surprise, the girls welcomed him into their social groups with warmth and without hesitation or judgment. Soon, Alex found himself surrounded by a group of kind and caring female friends who genuinely wanted the best for him and accepted him for who he was.  Unfortunately, while things were improving on one front, Alex still faced challenges from his male classmates. The other boys consistently mocked Alex for spending time with girls rather than joining them, leading to some humiliating moments, especially during sports day or lunch, where his exclusion was evident to all. Despite this, Alex remained determined not to let anything stand in the way of his newfound friendships with these brave young women, no matter what challenges came his way. |

Figure 5.3.2.7: Generated Story: Davinci – 2

In the narrative outlined in Section 5.3.1, the story was relatively simple, centered on a single character's experience with swimming and winning a medal. However, the more intricate plot introduced in Figure 5.3.2.1 involves Alex, the protagonist, who engages with two distinct groups in his class, each having its own actions and motivations. This added complexity presents a greater challenge for the generation models.

Figure 5.3.2.2 shows that the GPT2 model again generates incoherent text, deviating from the outline and extending the story beyond its intended scope. Similarly, the Bloomz-1B7 and Ada models, as depicted in Figures 5.3.2.3 and 5.3.2.4, do not show substantial improvement compared to GPT2.

Figure 5.3.2.5 presents the output from the Babbage model. While it largely rephrases the outline, it introduces a logical inconsistency in the final line, diverging from the intended plot.

Figure 5.3.2.6 displays the output from the Curie model, which offers a better performance by elaborating on the outline, although it misses a crucial plot point.

In contrast, Figure 5.3.2.7 illustrates the Davinci model's superior performance. It effectively captures the essence of the outline, producing a coherent and engaging narrative that aligns well with the intended storyline.

## **5.4 Summary**

Based on the results across various models, it is clear that the Davinci model consistently stands out by generating high-quality stories that align well with the provided outlines. In contrast, other models display a range of limitations, including generating incoherent narratives, merely rephrasing the outline without adding significant new content, or missing critical plot elements.

Moreover, it is evident that relying solely on quantitative metrics might not fully capture the intricate quality of generated stories. While these metrics provide useful insights into aspects such as fluency, coherence, and lexical overlap, they might miss key storytelling elements like creativity, narrative depth, and adherence to the prompt. Thus, qualitative evaluation, which incorporates human judgment and interpretation, is crucial for a thorough assessment of a model's storytelling capabilities.

# **CHAPTER 6**

# **CONCLUSION & RECOMMENDATIONS**

## **6.1 Discussion and Final Thoughts**

Throughout this thesis, several important conclusions have been drawn:

### **6.1.1 Optimal Model for Story Generation**

As illustrated in the examples from Section 5.3, larger language models tend to deliver superior results compared to smaller ones. For example, the Davinci model generates narratives that closely mimic human writing, even without further training, underscoring the significance of model size. However, it's also essential to consider the impact of instruction tuning. This factor plays a significant role in improving the model's ability to understand and follow the given prompts, particularly for enhancing the performance of models like Davinci.

### **6.1.2 Metrics for Evaluating Story Generation**

Another significant observation from comparing Sections 5.2 and 5.3 is the limitation of traditional quantitative metrics in evaluating high-quality narratives. While the stories produced by the Davinci model exhibit clear qualitative superiority, conventional metrics fail to capture this effectively. These metrics, designed primarily for tasks such as machine translation and summarization, are not always suitable for assessing open-ended story generation. This highlights the need for alternative evaluation metrics that are specifically tailored to assess narrative quality in such contexts.

## **6.2 Contribution**

## This thesis contributes to the field in several key ways:

* It sets benchmark scores for story generation tasks using a variety of open-source models and commercial APIs.
* It offers notebooks and code for data preprocessing, story generation, and evaluation, thereby improving reproducibility and supporting ongoing research in this domain.

Top of Form

Bottom of Form

## **6.3 Future Directions**

### **6.3.1 Comparison with Other Models**

While the OpenAI models demonstrate notable performance, it's crucial to acknowledge that this advantage may partly stem from the smaller size of the open-source models used for comparison. For example, GPT-2 models are relatively dated, and while BloomZ models incorporate advanced instruction tuning, they are not among the most cutting-edge models currently available. GPU memory limitations restricted the experiments to smaller models, so future research could benefit from exploring the performance of larger models and integrating them into benchmarking processes.

Additionally, this thesis focused solely on GPT-2 and BloomZ models from the open-source domain, leaving out other significant models such as Meta's OPT and OPT-IML. Comparing the BloomZ model with its previous versions could also shed light on its progress. Moreover, with the advent of OpenAI's ChatGPT framework, which builds on the InstructGPT approach with further enhancements for more fluid conversations and story generation, including ChatGPT in future benchmarking could provide valuable insights.

### **6.3.2 Comparison with Existing Methods**

This thesis presents a comparative evaluation of story generation results from different models, shedding light on their performance. However, it's crucial to acknowledge that similar approaches have been investigated in earlier research, such as the studies by Sun et al. (2020) and Fang et al. (2021). Future studies could improve the validation and contextual understanding of these findings by contrasting them with results from these previous investigations.

### **6.3.3 Improved Evaluation Metrics**

This thesis underscores a significant limitation of traditional metrics used for evaluating text generation tasks, particularly in the context of open-world story generation. Furthermore, these metrics often fall short in capturing critical narrative elements, such as creativity, suspense, and coherence.

To address these issues, recent research has introduced specialized benchmarks such as OpenMEVA (Guan et al., 2021) and HANNA (Chhun et al., 2022), which are specifically designed for evaluating story generation tasks. Future research could benefit from exploring the use of these specialized benchmarks to provide a more accurate assessment of story generation models.

# **REFERENCES**

Aghajanyan, A., Okhonko, D., Lewis, M., Joshi, M., Xu, H., Ghosh, G. and Zettlemoyer, L., (2021) HTLM: Hyper-Text Pre-Training and Prompting of Language Models. [online] Available at: <https://arxiv.org/abs/2107.06955v1> [Accessed 10 Feb. 2023].

Alhussain, A.I. and Azmi, A.M., (2021) Automatic Story Generation. *ACM Computing Surveys (CSUR)*, [online] 54(5), pp.1-38. Available at: <https://dl.acm.org/doi/10.1145/3453156> [Accessed 7 Oct. 2022].

Ammanabrolu, P., Tien, E., Cheung, W., Luo, Z., Ma, W., Martin, L.J. and Riedl, M.O., (2019) Guided Neural Language Generation for Automated Storytelling. [online] pp.46–55. Available at: <https://aclanthology.org/W19-3405> [Accessed 5 Jan. 2023].

Ammanabrolu, P., Tien, E., Cheung, W., Luo, Z., Ma, W., Martin, L.J. and Riedl, M.O., (2020) Story Realization: Expanding Plot Events into Sentences. *Proceedings of the AAAI Conference on Artificial Intelligence*, [online] 3405, pp.7375–7382. Available at: <https://ojs.aaai.org/index.php/AAAI/article/view/6232> [Accessed 5 Jan. 2023].

Araz, K., (2020) Transformer Neural Networks for Automated Story Generation. *Dissertations*. [online] Available at: <https://arrow.tudublin.ie/scschcomdis/211> [Accessed 5 Jan. 2023].

Bach, S.H., Sanh, V., Yong, Z.-X., Webson, A., Raffel, C., Nayak, N. V., Sharma, A., Kim, T., Bari, M.S., Fevry, T., Alyafeai, Z., Dey, M., Santilli, A., Sun, Z., Ben-David, S., Xu, C., Chhablani, G., Wang, H., Fries, J.A., Al-shaibani, M.S., Sharma, S., Thakker, U., Almubarak, K., Tang, X., Radev, D., Jiang, M.T.-J. and Rush, A.M., (2022) PromptSource: An Integrated Development Environment and Repository for Natural Language Prompts. [online] pp.93–104. Available at: <https://arxiv.org/abs/2202.01279v3> [Accessed 16 Feb. 2023].

Bahdanau, D., Cho, K.H. and Bengio, Y., (2014) Neural Machine Translation by Jointly Learning to Align and Translate. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. [online] Available at: <https://arxiv.org/abs/1409.0473v7> [Accessed 12 Dec. 2022].

Bakhtin, A., Deng, Y., Ott, M., Ranzato, M.’ A. and Szlam, A., (2021) Residual Energy-Based Models for Text. *Journal of Machine Learning Research*, [online] 22, pp.1–41. Available at: <http://jmlr.org/papers/v22/20-326.html>. [Accessed 23 Oct. 2022].

Balasubramanian, N., Soderland, S. and Etzioni, O., (2012) Rel-grams: A Probabilistic Model of Relations in Text. pp.101–105.

Bao, H., Dong, L., Wei, F., Wang, W., Yang, N., Liu, X., Wang, Y., Gao, J., Piao, S., Zhou, M. and Hon, H.-W., (2020) *UniLMv2: Pseudo-Masked Language Models for Unified Language Model Pre-Training*. pp. 642-652. Available at: <https://proceedings.mlr.press/v119/bao20a.html> [Accessed 10 Feb. 2023].

Baumgartner, J., Zannettou, S., Keegan, B., Squire, M. and Blackburn, J., (2020) The Pushshift Reddit Dataset. Proceedings of the International AAAI Conference on Web and Social Media, [online] 14, pp.830–839. Available at: <https://ojs.aaai.org/index.php/ICWSM/article/view/7347> [Accessed 16 Feb. 2023].

Behrooz, M., Robertson, J. and Jhala, A., (2019) Story Quality as a Matter of Perception: Using Word Embeddings to Estimate Cognitive Interest. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, [online] 151, pp.3–9. Available at: <https://ojs.aaai.org/index.php/AIIDE/article/view/5217> [Accessed 5 Jan. 2023].

Ben-David, E., Oved, N. and Reichart, R., (2021) PADA: Example-based Prompt Learning for on-the-fly Adaptation to Unseen Domains. *Transactions of the Association for Computational Linguistics*, [online] 10, pp.414–433. Available at: <https://arxiv.org/abs/2102.12206v4> [Accessed 10 Feb. 2023].

Bisk, Y., Holtzman, A., Thomason, J., Andreas, J., Bengio, Y., Chai, J., Lapata, M., Lazaridou, A., May, J., Nisnevich, A., Pinto, N. and Turian, J., (2020) Experience Grounds Language. *EMNLP 2020 - 2020 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, [online] pp.8718–8735. Available at: <https://arxiv.org/abs/2004.10151v3> [Accessed 23 Oct. 2022].

Black, J.B. and Bower, G.H., (1980) Story understanding as problem-solving. *Poetics*, 91–3, pp.223–250.

Black, J.B. and Wilensky, R., (1979) An Evaluation of Story Grammars\*. *Cognitive Science*, [online] 33, pp.213–229. Available at: <https://onlinelibrary.wiley.com/doi/full/10.1207/s15516709cog0303_2> [Accessed 5 Jan. 2023].

Black, S., Gao, L., Wang, P., Leahy, C. and Biderman, S., (2021) GPT-Neo: Large scale autoregressive language modeling with mesh tensorflow. [online] Available at: <https://zenodo.org/record/5551208> [Accessed 13 Feb. 2023].

Bosselut, A., Celikyilmaz, A., He, X., Gao, J., Huang, P. sen and Choi, Y., (2018) Discourse-Aware Neural Rewards for Coherent Text Generation. *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, [online] 1, pp.173–184. Available at: <https://arxiv.org/abs/1805.03766v1> [Accessed 23 Oct. 2022].

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. *Advances in Neural Information Processing Systems*, [online] 33, pp.1877–1901. Available at: <https://commoncrawl.org/the-data/> [Accessed 10 Feb. 2023].

Chai, D., Wu, W., Han, Q., Wu, F. and Li, J., (2020) *Description Based Text Classification with Reinforcement Learning, pp. 1371-1382. PMLR*. Available at: <https://proceedings.mlr.press/v119/chai20a.html> [Accessed 10 Feb. 2023].

Chambers, N. and Jurafsky, D., (2008) Unsupervised Learning of Narrative Event Chains. *In Proceedings of ACL-08: HLT* (pp. 789-797).

Chambers, N. and Jurafsky, D., (2009) Unsupervised Learning of Narrative Schemas and their Participants. *ACL and AFNLP*, pp.602–610.

Chambers, N., Wang, S. and Jurafsky, D., (2007) Classifying Temporal Relations Between Events. [online] pp.173–176. Available at: <http://nlp.stanford.edu/software/lex-parser.shtml> [Accessed 5 Jan. 2023].

Chaturvedi, S., Peng, H. and Roth, D., (2017) Story Comprehension for Predicting What Happens Next. *EMNLP 2017 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, [online] pp.1603–1614. Available at: <https://aclanthology.org/D17-1168> [Accessed 5 Jan. 2023].

Chen, G., Liu, Y., Luan, H., Zhang, M., Liu, Q. and Sun, M., (2021) Learning to Generate Explainable Plots for Neural Story Generation. *IEEE/ACM Transactions on Audio Speech and Language Processing*, 29, pp.585–593.

Chen, J., Chen, J. and Yu, Z., (2019) Incorporating Structured Commonsense Knowledge in Story Completion. *Proceedings of the AAAI Conference on Artificial Intelligence*, [online] 3301, pp.6244–6251. Available at: <https://ojs.aaai.org/index.php/AAAI/article/view/5183> [Accessed 5 Jan. 2023].

Chhun, C., Colombo, P., Clavel, C. and Suchanek, F.M., (2022) Of Human Criteria and Automatic Metrics: A Benchmark of the Evaluation of Story Generation. [online] Available at: <https://arxiv.org/abs/2208.11646v4> [Accessed 13 Feb. 2023].

Choi, Y., Kim, S. and Lee, J.H., (2016) Recurrent Neural Network for Storytelling. *Proceedings - 2016 Joint 8th International Conference on Soft Computing and Intelligent Systems and 2016 17th International Symposium on Advanced Intelligent Systems, SCIS-ISIS 2016*, pp.841–845.

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H.W., Sutton, C., Gehrmann, S., Schuh, P., Shi, K., Tsvyashchenko, S., Maynez, J., Rao, A., Barnes, P., Tay, Y., Shazeer, N., Prabhakaran, V., Reif, E., Du, N., Hutchinson, B., Pope, R., Bradbury, J., Austin, J., Isard, M., Gur-Ari, G., Yin, P., Duke, T., Levskaya, A., Ghemawat, S.

Christiano, P.F., Leike, J., Brown, T.B., Martic, M., Legg, S. and Amodei, D., (2017) Deep reinforcement learning from human preferences. *Advances in Neural Information Processing Systems*, [online] 2017-December, pp.4300–4308. Available at: <https://arxiv.org/abs/1706.03741v3> [Accessed 16 Feb. 2023].

Chu, C., Dabre, R. and Kurohashi, S., (2017) An Empirical Comparison of Domain Adaptation Methods for Neural Machine Translation. *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, [online] 2, pp.385–391. Available at: <https://aclanthology.org/P17-2061> [Accessed 10 Feb. 2023].

Chung, J., Gulcehre, C., Cho, K. and Bengio, Y., (2014) Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. [online] Available at: <https://arxiv.org/abs/1412.3555v1> [Accessed 12 Dec. 2022].

Clark, E., August, T., Serrano, S., Haduong, N., Gururangan, S. and Smith, N.A., (2021) All That’s ‘Human’ Is Not Gold: Evaluating Human Evaluation of Generated Text. *ACL-IJCNLP 2021 - 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, [online] pp.7282–7296. Available at: <https://arxiv.org/abs/2107.00061v2> [Accessed 23 Oct. 2022].

Collobert, R., Weston, J., Com, J., Karlen, M., Kavukcuoglu, K. and Kuksa, P., (2011) Natural Language Processing (Almost) from Scratch. *Journal of Machine Learning Research*, 12, pp.2493–2537.

Commoncrawl.org, (2023) *So you’re ready to get started. – Common Crawl*. [online] Available at: <https://commoncrawl.org/the-data/get-started/> [Accessed 16 Feb. 2023].

Cook, W.W., (2011) Plotto : the master book of all plots. p.40.

Cui, L., Wu, Y., Liu, J., Yang, S. and Zhang, Y., (2021) Template-Based Named Entity Recognition Using BART. *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, [online] pp.1835–1845. Available at: <https://arxiv.org/abs/2106.01760v1> [Accessed 10 Feb. 2023].

cs.rochester.edu, (2022) *ROCStories and the Story Cloze Test*. [online] Available at: <https://cs.rochester.edu/nlp/rocstories/> [Accessed 26 Oct. 2022].

Devlin, J., Chang, M.W., Lee, K. and Toutanova, K., (2019) BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, [online] 1, pp.4171–4186. Available at: <https://arxiv.org/abs/1810.04805v2> [Accessed 10 Feb. 2023].

Doddapaneni, S., Ramesh, G., Khapra, M.M., Kunchukuttan, A., Kumar, P. and Madras, I., (2021) A Primer on Pretrained Multilingual Language Models. [online] Available at: <https://arxiv.org/abs/2107.00676v2> [Accessed 10 Feb. 2023].

Dodge, J., Ilharco, G., Schwartz, R., Farhadi, A., Hajishirzi, H. and Smith, N., (2020) Fine-Tuning Pretrained Language Models: Weight Initializations, Data Orders, and Early Stopping. [online] Available at: <https://arxiv.org/abs/2002.06305v1> [Accessed 10 Feb. 2023].

Dong, L., Yang, N., Wang, W., Wei, F., Liu, X., Wang, Y., Gao, J., Zhou, M. and Hon, H.-W., (2019a) Unified Language Model Pre-training for Natural Language Understanding and Generation. *Advances in Neural Information Processing Systems*, [online] 32. Available at: <https://github.com/microsoft/unilm>. [Accessed 12 Dec. 2022].

Dong, L., Yang, N., Wang, W., Wei, F., Liu, X., Wang, Y., Gao, J., Zhou, M. and Hon, H.-W., (2019b) Unified Language Model Pre-training for Natural Language Understanding and Generation. *Advances in Neural Information Processing Systems*, [online] 32. Available at: <https://github.com/microsoft/unilm>. [Accessed 10 Feb. 2023].

Dou, Y., Forbes, M., Koncel-Kedziorski, R., Smith, N.A. and Choi, Y., (2021) Is GPT-3 Text Indistinguishable from Human Text? Scarecrow: A Framework for Scrutinizing Machine Text. [online] pp.7250–7274. Available at: <https://arxiv.org/abs/2107.01294v3> [Accessed 23 Oct. 2022].

Dou, Z.Y., Liu, P., Hayashi, H., Jiang, Z. and Neubig, G., (2020) GSum: A General Framework for Guided Neural Abstractive Summarization. *NAACL-HLT 2021 - 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference*, [online] pp.4830–4842. Available at: <https://arxiv.org/abs/2010.08014v3> [Accessed 10 Feb. 2023].

Dziri, N., Madotto, A., Zaiane, O. and Bose, A.J., (2021) Neural Path Hunter: Reducing Hallucination in Dialogue Systems via Path Grounding. *EMNLP 2021 - 2021 Conference on Empirical Methods in Natural Language Processing, Proceedings*, [online] pp.2197–2214. Available at: <https://arxiv.org/abs/2104.08455v2> [Accessed 23 Oct. 2022].

Eger, M., Potts, C.M., Barot, C. and Young, R.M., (2015) Plotter: Operationalizing the Master Book of All Plots. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, [online] 114, pp.30–33. Available at: <https://ojs.aaai.org/index.php/AIIDE/article/view/12835> [Accessed 9 Feb. 2023].

Fadaee, M., Bisazza, A. and Monz, C., (2017) Data Augmentation for Low-Resource Neural Machine Translation. *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, [online] 2, pp.567–573. Available at: <http://arxiv.org/abs/1705.00440> [Accessed 10 Feb. 2023].

Fan, A., Lewis, M. and Dauphin, Y., (2018) Hierarchical Neural Story Generation. *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics.* Available at: <https://arxiv.org/abs/1805.04833v1> [Accessed 9 Feb. 2023].

Fan, A., Lewis, M. and Dauphin, Y., (2019) Strategies for Structuring Story Generation. *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, [online] pp.2650–2660. Available at: <https://arxiv.org/abs/1902.01109v2> [Accessed 9 Feb. 2023].

Fang, L., Zeng, T., Liu, C., Bo, L., Dong, W. and Chen, C., (2021) Outline to Story: Fine-grained Controllable Story Generation from Cascaded Events. Available at: <https://arxiv.org/abs/2101.00822>

Finn, C., Abbeel, P. and Levine, S., (2017) Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. *In International conference on machine learning* (pp. 1126-1135). PMLR. Available at: <https://proceedings.mlr.press/v70/finn17a.html> [Accessed 10 Feb. 2023].

Gao, J., Peng, B., Li, C., Li, J., Shayandeh, S., Liden, L. and Shum, H.-Y., (2020a) Robust Conversational AI with Grounded Text Generation. [online] Available at: <https://arxiv.org/abs/2009.03457v1> [Accessed 23 Oct. 2022].

Gao, L., Biderman, S., Black, S., Golding, L., Hoppe, T., Foster, C., Phang, J., He, H., Thite, A., Nabeshima, N., Presser, S. and Leahy, C., (2020b) The Pile: An 800GB Dataset of Diverse Text for Language Modeling. [online] Available at: <https://arxiv.org/abs/2101.00027v1> [Accessed 16 Feb. 2023].

Gao, T., Fisch, A. and Chen, D., (2020c) Making Pre-trained Language Models Better Few-shot Learners. *ACL-IJCNLP 2021 - 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, [online] pp.3816–3830. Available at: <https://arxiv.org/abs/2012.15723v2> [Accessed 10 Feb. 2023].

Gervás, P., (2009) Computational Approaches to Storytelling and Creativity. *AI Magazine*, [online] 303, pp.49–49. Available at: <https://ojs.aaai.org/aimagazine/index.php/aimagazine/article/view/2250> [Accessed 9 Feb. 2023].

Goldfarb-Tarrant, S., Chakrabarty, T., Weischedel, R. and Peng, N., (2020) Content Planning for Neural Story Generation with Aristotelian Rescoring. *EMNLP 2020 - 2020 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, [online] pp.4319–4338. Available at: <https://arxiv.org/abs/2009.09870v2> [Accessed 23 Oct. 2022].

Grangier, D. and Auli, M., (2017) QuickEdit: Editing Text & Translations by Crossing Words Out. *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, [online] 1, pp.272–282. Available at: <https://arxiv.org/abs/1711.04805v2> [Accessed 10 Feb. 2023].

Guan, J., Huang, F., Zhao, Z., Zhu, X. and Huang, M., (2020) A Knowledge-Enhanced Pretraining Model for Commonsense Story Generation. *Transactions of the Association for Computational Linguistics*, [online] 8, pp.93–108. Available at: <https://aclanthology.org/2020.tacl-1.7> [Accessed 31 Aug. 2022].

Guan, J., Wang, Y. and Huang, M., (2019) Story Ending Generation with Incremental Encoding and Commonsense Knowledge. *Proceedings of the AAAI Conference on Artificial Intelligence*, [online] 3301, pp.6473–6480. Available at: <https://ojs.aaai.org/index.php/AAAI/article/view/4612> [Accessed 9 Feb. 2023].

Guan, J., Zhang, Z., Feng, Z., Liu, Z., Ding, W., Mao, X., Fan, C. and Huang, M., (2021) OpenMEVA: A Benchmark for Evaluating Open-ended Story Generation Metrics. *ACL-IJCNLP 2021 - 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, [online] pp.6394–6407. Available at: <https://arxiv.org/abs/2105.08920v1> [Accessed 13 Feb. 2023].

Guyon, I., Weston, J. and Barnhill, S., (2002) Gene Selection for Cancer Classification using Support Vector Machines. 46, pp.389–422.

Hambardzumyan, K., Khachatrian, H. and May, J., (2021) WARP: Word-level Adversarial ReProgramming. *ACL-IJCNLP 2021 - 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, [online] pp.4921–4933. Available at: <https://arxiv.org/abs/2101.00121v2> [Accessed 10 Feb. 2023].

Han, X., Zhao, W., Ding, N., Liu, Z. and Sun, M., (2022) PTR: Prompt Tuning with Rules for Text Classification. *AI Open*, 3, pp.182–192.

Harrison, B., Purdy, C. and Riedl, M.O., (2017) Toward Automated Story Generation with Markov Chain Monte Carlo Methods and Deep Neural Networks. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment* (Vol. 13, No. 2, pp. 191-197). [online] Available at: www.aaai.org [Accessed 12 Feb. 2023].

Hochreiter, S. and Schmidhuber, J., (1997) Long Short-Term Memory. *Neural Computation*, 98, pp.1735–1780.

Holtzman, A., Buys, J., Du, L., Forbes, M. and Choi, Y., (2019) The Curious Case of Neural Text Degeneration. *CEUR Workshop Proceedings*, [online] 2540. Available at: <https://arxiv.org/abs/1904.09751v2> [Accessed 9 Feb. 2023].

Hu, L., Li, J., Nie, L., Li, X.L. and Shao, C., (2017) What Happens Next? Future Subevent Prediction Using Contextual Hierarchical LSTM. *Proceedings of the AAAI Conference on Artificial Intelligence*, [online] 311, pp.3450–3456. Available at: <https://ojs.aaai.org/index.php/AAAI/article/view/11001> [Accessed 9 Feb. 2023].

Ippolito, D., Grangier, D., Eck, D. and Callison-Burch, C., (2020) Toward Better Storylines with Sentence-Level Language Models. [online] pp.7472–7478. Available at: <https://arxiv.org/abs/2005.05255v1> [Accessed 9 Feb. 2023].

Iyer, S., Lin, X.V., Pasunuru, R., Mihaylov, T., Simig, D., Yu, P., Shuster, K., Wang, T., Liu, Q., Koura, P.S., Li, X., O’Horo, B., Pereyra, G., Wang, J., Dewan, C., Celikyilmaz, A., Zettlemoyer, L. and Stoyanov, V., (2022) OPT-IML: Scaling Language Model Instruction Meta Learning through the Lens of Generalization. [online] Available at: <https://arxiv.org/abs/2212.12017v3> [Accessed 16 Feb. 2023].

Jain, P., Agrawal, P., Mishra, A., Sukhwani, M., Laha, A. and Sankaranarayanan, K., (2017) Story Generation from Sequence of Independent Short Descriptions. [online] 7. Available at: <https://arxiv.org/abs/1707.05501v2> [Accessed 9 Feb. 2023].

Jans, B., Bethard, S., Vulic, I., Moens, M.-F., Daelemans, W., Lapata, M. and Màrquez, L., (2012) Skip n-grams and ranking functions for predicting script events. *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2012*), (pp. 336-344). ACL. [online] Available at: <https://lirias.kuleuven.be/1572145> [Accessed 9 Feb. 2023].

Ji, H. and Huang, M., (2021) DiscoDVT: Generating Long Text with Discourse-Aware Discrete Variational Transformer. *EMNLP 2021 - 2021 Conference on Empirical Methods in Natural Language Processing, Proceedings*, [online] pp.4208–4224. Available at: <https://arxiv.org/abs/2110.05999v1> [Accessed 23 Oct. 2022].

Ji, H., Ke, P., Huang, S., Wei, F., Zhu, X. and Huang, M., (2020) Language Generation with Multi-Hop Reasoning on Commonsense Knowledge Graph. *EMNLP 2020 - 2020 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, [online] pp.725–736. Available at: <https://arxiv.org/abs/2009.11692v1> [Accessed 23 Oct. 2022].

Jiang, Z., Anastasopoulos, A., Araki, J., Ding, H. and Neubig, G., (2020a) X-FACTR: Multilingual Factual Knowledge Retrieval from Pretrained Language Models. *EMNLP 2020 - 2020 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, [online] pp.5943–5959. Available at: <https://arxiv.org/abs/2010.06189v3> [Accessed 10 Feb. 2023].

Jiang, Z., Xu, F.F., Araki, J. and Neubig, G., (2020b) How can we know what language models know? *Transactions of the Association for Computational Linguistics*, [online] 8, pp.423–438. Available at: <https://direct.mit.edu/tacl/article/doi/10.1162/tacl_a_00324/96460/How-Can-We-Know-What-Language-Models-Know> [Accessed 10 Feb. 2023].

Jurafsky, D. and Martin, J.H., (2000) Book Reviews Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. [online] Available at: www.cs.colorado.edu/~martin/slp.html. [Accessed 10 Feb. 2023].

Kaiser, Ł., Brain, G., Nachum, O., Google Brain, †, Roy, A., Tech, G. and Bengio, S., (2017) Learning to Remember Rare Events. *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*. [online] Available at: <https://arxiv.org/abs/1703.03129v1> [Accessed 10 Feb. 2023].

Kalchbrenner, N., Grefenstette, E. and Blunsom, P., (2014) A Convolutional Neural Network for Modelling Sentences. *52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014 - Proceedings of the Conference*, [online] 1, pp.655–665. Available at: <https://arxiv.org/abs/1404.2188v1> [Accessed 12 Dec. 2022].

Kartal, B., Koenig, J. and Guy, S.J., (2014) User-Driven Narrative Variation in Large Story Domains using Monte Carlo Tree Search. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*. [online], pp. 69-76. Available at: www.ifaamas.org [Accessed 9 Feb. 2023].

Keskar, N.S., Mccann, B., Varshney, L.R., Xiong, C., Socher, R. and Research, S., (2019) CTRL: A Conditional Transformer Language Model for Controllable Generation. [online] Available at: <https://arxiv.org/abs/1909.05858v2> [Accessed 23 Oct. 2022].

Khandelwal, U., He, H., Qi, P. and Jurafsky, D., (2018) Sharp Nearby, Fuzzy Far Away: How Neural Language Models Use Context. *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, [online] 1, pp.284–294. Available at: <https://arxiv.org/abs/1805.04623v1> [Accessed 9 Feb. 2023].

Khashabi, D., Min, S., Khot, T., Sabharwal, A., Tafjord, O., Clark, P. and Hajishirzi, H., (2020) UnifiedQA: Crossing Format Boundaries With a Single QA System. *Findings of the Association for Computational Linguistics Findings of ACL: EMNLP 2020*, [online] pp.1896–1907. Available at: <https://arxiv.org/abs/2005.00700v3> [Accessed 10 Feb. 2023].

Kikuchi, Y., Neubig, G., Sasano, R., Takamura, H. and Okumura, M., (2016) Controlling Output Length in Neural Encoder-Decoders. *EMNLP 2016 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, [online] pp.1328–1338. Available at: <https://arxiv.org/abs/1609.09552v1> [Accessed 10 Feb. 2023].

Kotsiantis, S.B., (2007) Supervised Machine Learning: A Review of Classification Techniques. *Informatica*, 31, pp.249–268.

Kumar, A., Irsoy, O., Ondruska, P., Iyyer, M., Bradbury, J., Gulrajani, I., Zhong, V., Paulus, R. and Socher, R., (2016) Ask Me Anything: Dynamic Memory Networks for Natural Language Processing. In *International conference on machine learning* (pp. 1378-1387). Available at: <https://proceedings.mlr.press/v48/kumar16.html> [Accessed 10 Feb. 2023].

Kumar, S. and Talukdar, P., (2021) Reordering Examples Helps during Priming-based Few-Shot Learning. *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, [online] pp.4507–4518. Available at: <https://arxiv.org/abs/2106.01751v1> [Accessed 10 Feb. 2023].

Kybartas, B. and Bidarra, R., (2017) A Survey on Story Generation Techniques for Authoring Computational Narratives. *IEEE Transactions on Computational Intelligence and AI in Games*, 93, pp.239–253.

Lafferty, J., McCallum, A. and Pereira, F., (2001) Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. *Departmental Papers (CIS)*. [online] Available at: <https://repository.upenn.edu/cis_papers/159> [Accessed 12 Dec. 2022].

León, C. and Gervás, P., (2010) The role of evaluation-driven rejection in the successful exploration of a conceptual space of stories. *Minds and Machines*, [online] 204, pp.615–634. Available at: <https://link.springer.com/article/10.1007/s11023-010-9205-z> [Accessed 9 Feb. 2023].

Lester, B., Al-Rfou, R. and Constant, N., (2021) The Power of Scale for Parameter-Efficient Prompt Tuning. *EMNLP 2021 - 2021 Conference on Empirical Methods in Natural Language Processing, Proceedings*, [online] pp.3045–3059. Available at: <https://arxiv.org/abs/2104.08691v2> [Accessed 10 Feb. 2023].

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L. and Ai, F., (2020) BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. [online] pp.7871–7880. Available at: <https://aclanthology.org/2020.acl-main.703> [Accessed 13 Feb. 2023].

Li, B., Lee-Urban, S., Johnston, G. and Riedl, M.O., (2013) Story Generation with Crowdsourced Plot Graphs. *Proceedings of the AAAI Conference on Artificial Intelligence*, [online] 271, pp.598–604. Available at: <https://ojs.aaai.org/index.php/AAAI/article/view/8649> [Accessed 9 Feb. 2023].

Li, Q., Li, Z., Wei, J.-M., (顾彦慧) Y.G., Jatowt, A. and Yang, Z., (2018a) A Multi-Attention based Neural Network with External Knowledge for Story Ending Predicting Task. In *Proceedings of the 27th International Conference on Computational Linguistics* (pp. 1754-1762). Available at: <https://aclanthology.org/C18-1149> [Accessed 9 Feb. 2023].

Li, X.L. and Liang, P., (2021) Prefix-Tuning: Optimizing Continuous Prompts for Generation. *ACL-IJCNLP 2021 - 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, [online] pp.4582–4597. Available at: <https://arxiv.org/abs/2101.00190v1> [Accessed 10 Feb. 2023].

Li, Z., Ding, X. and Liu, T., (2018b) Generating Reasonable and Diversified Story Ending Using Sequence to Sequence Model with Adversarial Training. In *Proceedings of the 27th International Conference on Computational Linguistics* (pp. 1033-1043). Available at: <https://aclanthology.org/C18-1088> [Accessed 9 Feb. 2023].

Li, Z., Ding, X., Liu, T., Hu, J.E. and van Durme, B., (2021) Guided Generation of Cause and Effect. *IJCAI International Joint Conference on Artificial Intelligence*, [online] 2021-January, pp.3629–3636. Available at: <http://arxiv.org/abs/2107.09846> [Accessed 9 Feb. 2023].

Liu, C.W., Lowe, R., Serban, I. v., Noseworthy, M., Charlin, L. and Pineau, J., (2016) How NOT To Evaluate Your Dialogue System: An Empirical Study of Unsupervised Evaluation Metrics for Dialogue Response Generation. *EMNLP 2016 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, [online] pp.2122–2132. Available at: <https://arxiv.org/abs/1603.08023v2> [Accessed 9 Feb. 2023].

Lin, C.-Y., (2004) ROUGE: A Package for Automatic Evaluation of Summaries. In *Text summarization branches out* (pp. 74-81). Available at: <https://aclanthology.org/W04-1013> [Accessed 29 Oct. 2022].

Liu, J., Shen, D., Zhang, Y., Dolan, B., Carin, L. and Chen, W., (2021a) What Makes Good In-Context Examples for GPT-$3$? *DeeLIO 2022 - Deep Learning Inside Out: 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures, Proceedings of the Workshop*, [online] pp.100–114. Available at: <https://arxiv.org/abs/2101.06804v1> [Accessed 10 Feb. 2023].

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H. and Neubig, G., (2021b) Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. *ACM Computing Surveys*, , 55(9), pp.1-35.. [online] Available at: <https://arxiv.org/abs/2107.13586v1> [Accessed 10 Nov. 2022].

Liu, X., Zheng, Y., Du, Z., Ding, M., Qian, Y., Yang, Z. and Tang, J., (2021c) GPT Understands, Too. [online] Available at: <https://arxiv.org/abs/2103.10385v1> [Accessed 10 Feb. 2023].

Liu, Y., Gu, J., Goyal, N., Li, X., Edunov, S., Ghazvininejad, M., Lewis, M. and Zettlemoyer, L., (2020) Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, [online] 8, pp.726–742. Available at: <https://direct.mit.edu/tacl/article/doi/10.1162/tacl_a_00343/96484/Multilingual-Denoising-Pre-training-for-Neural> [Accessed 10 Feb. 2023].

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V. and Allen, P.G., (2019) RoBERTa: A Robustly Optimized BERT Pretraining Approach. [online] Available at: <https://arxiv.org/abs/1907.11692v1> [Accessed 10 Feb. 2023].

Liu, R. and Nikolic, P.K., (2021) Mutltimodal AI Companion for Interactive Fairytale Co-creation. [online] Available at: <https://arxiv.org/abs/2112.00331v1> [Accessed 6 Sep. 2022].

Logan Iv, R.L., Balaževi´cbalaževi´c, I., Wallace, E., Petroni, F., Singh, S. and Riedel, S., (2021) Cutting Down on Prompts and Parameters: Simple Few-Shot Learning with Language Models. [online] pp.2824–2835. Available at: <https://arxiv.org/abs/2106.13353v2> [Accessed 10 Feb. 2023].

Mark, G.W. and Clark, S., (2016) What Happens Next? Event Prediction Using a Compositional Neural Network Model. *Proceedings of the AAAI Conference on Artificial Intelligence*, [online] 301, pp.2727–2733. Available at: <https://ojs.aaai.org/index.php/AAAI/article/view/10344> [Accessed 9 Feb. 2023].

Martin, L.J., Ammanabrolu, P., Wang, X., Hancock, W., Singh, S., Harrison, B. and Riedl, M.O., (2018) Event Representations for Automated Story Generation with Deep Neural Nets. *Proceedings of the AAAI Conference on Artificial Intelligence*, [online] 321. Available at: <https://ojs.aaai.org/index.php/AAAI/article/view/11430> [Accessed 9 Feb. 2023].

McCann, B., Keskar, N.S., Xiong, C. and Socher, R., (2018) The Natural Language Decathlon: Multitask Learning as Question Answering. [online] Available at: <https://arxiv.org/abs/1806.08730v1> [Accessed 10 Feb. 2023].

McCloskey, M. and Cohen, N.J., (1989) Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. *Psychology of Learning and Motivation - Advances in Research and Theory*, 24C, pp.109–165.

Mcintyre, N. and Lapata, M., (2010) Plot Induction and Evolutionary Search for Story Generation. [online] pp.11–16. Available at: <http://opennlp.sourceforge.net/>. [Accessed 9 Feb. 2023].

Mostafazadeh, N., Chambers, N., He, X., Parikh, D., Batra, D., Vanderwende, L., Kohli, P. and Allen, J., (2016a) A Corpus and Cloze Evaluation for Deeper Understanding of Commonsense Stories. pp.839–849.

Mostafazadeh, N., Vanderwende, L., Yih, W.-T., Kohli, P. and Allen, J., (2016b) Story Cloze Evaluator: Vector Space Representation Evaluation by Predicting What Happens Next. [online] pp.24–29. Available at: <http://cs.rochester.edu/> [Accessed 9 Feb. 2023].

Muennighoff, N., Wang, T., Sutawika, L., Roberts, A., Biderman, S., Scao, T. Le, Bari, M.S., Shen, S., Yong, Z.-X., Schoelkopf, H., Tang, X., Radev, D., Aji, A.F., Almubarak, K., Albanie, S., Alyafeai, Z., Webson, A., Raff, E. and Raffel, C., (2022) Crosslingual Generalization through Multitask Finetuning. [online] Available at: <https://arxiv.org/abs/2211.01786v1> [Accessed 16 Feb. 2023].

Och, F.J. and Ney, H., (2004) The Alignment Template Approach to Statistical Machine Translation. *Computational Linguistics*, [online] 304, pp.417–449. Available at: <https://direct.mit.edu/coli/article/30/4/417/1869/The-Alignment-Template-Approach-to-Statistical> [Accessed 12 Dec. 2022].

O’Neill, B. and Riedl, M., (2014) Dramatis: A Computational Model of Suspense. *Proceedings of the AAAI Conference on Artificial Intelligence*, [online] 281, pp.944–950. Available at: <https://ojs.aaai.org/index.php/AAAI/article/view/8836> [Accessed 10 Feb. 2023].

OpenAI, (2023a) *ChatGPT: Optimizing Language Models for Dialogue*. [online] Available at: <https://openai.com/blog/chatgpt/> [Accessed 17 Feb. 2023].

OpenAI, (2023b) *Model index for researchers - OpenAI API*. [online] Available at: <https://platform.openai.com/docs/model-index-for-researchers> [Accessed 16 Feb. 2023].

Openai, A.R., Openai, K.N., Openai, T.S. and Openai, I.S., (2018) Improving Language Understanding by Generative Pre-Training. [online] Available at: <https://gluebenchmark.com/leaderboard> [Accessed 12 Dec. 2022].

Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., Grangier, D. and Auli, M., (2019) fairseq: A Fast, Extensible Toolkit for Sequence Modeling. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Demonstrations Session*, [online] pp.48–53. Available at: <https://arxiv.org/abs/1904.01038v1> [Accessed 10 Feb. 2023].

Ouyang, X., Wang, S., Pang, C., Sun, Y., Tian, H., Wu, H. and Wang, H., (2020) ERNIE-M: Enhanced Multilingual Representation by Aligning Cross-lingual Semantics with Monolingual Corpora. *EMNLP 2021 - 2021 Conference on Empirical Methods in Natural Language Processing, Proceedings*, [online] pp.27–38. Available at: <https://arxiv.org/abs/2012.15674v4> [Accessed 10 Feb. 2023].

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C.L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J. and Lowe, R., (2022) Training language models to follow instructions with human feedback. [online] Available at: <https://arxiv.org/abs/2203.02155v1> [Accessed 16 Feb. 2023].

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* (pp. 311-318).

Peinado, F. and Gervás, P., (2006) Evaluation of automatic generation of basic stories. *New Generation Computing*, [online] 243, pp.289–302. Available at: <https://link.springer.com/article/10.1007/BF03037336> [Accessed 10 Feb. 2023].

Perez, E., Kiela, D. and Cho, K., (2021) True Few-Shot Learning with Language Models. *Advances in Neural Information Processing Systems*, 34, pp.11054–11070.

PÉrez, R.Pé.Ý. and Sharples, M., (2010) MEXICA: A computer model of a cognitive account of creative writing. [*http://dx.doi.org/10.1080/09528130010029820*](http://dx.doi.org/10.1080/09528130010029820), [online] 132, pp.119–139. Available at: <https://www.tandfonline.com/doi/abs/10.1080/09528130010029820> [Accessed 10 Feb. 2023].

Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K. and Zettlemoyer, L., (2018) Deep Contextualized Word Representations. *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, [online] 1, pp.2227–2237. Available at: <https://aclanthology.org/N18-1202> [Accessed 12 Dec. 2022].

Peters, M.E., Ruder, S. and Smith, N.A., (2019) To Tune or Not to Tune? Adapting Pretrained Representations to Diverse Tasks. *ACL 2019 - 4th Workshop on Representation Learning for NLP, RepL4NLP 2019 - Proceedings of the Workshop*, [online] pp.7–14. Available at: <https://arxiv.org/abs/1903.05987v2> [Accessed 10 Feb. 2023].

Petroni, F., Rocktäschel, T., Lewis, P., Bakhtin, A., Wu, Y., Miller, A.H. and Riedel, S., (2019) Language Models as Knowledge Bases? *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, [online] pp.2463–2473. Available at: <https://arxiv.org/abs/1909.01066v2> [Accessed 10 Feb. 2023].

Pichotta, K. and Mooney, R.J., (2014) Statistical Script Learning with Multi-Argument Events. pp.220–229.

Pichotta, K. and Mooney, R.J., (2016a) Learning Statistical Scripts with LSTM Recurrent Neural Networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, [online] 301, pp.2800–2806. Available at: <https://ojs.aaai.org/index.php/AAAI/article/view/10347> [Accessed 10 Feb. 2023].

Pichotta, K. and Mooney, R.J., (2016b) Using Sentence-Level LSTM Language Models for Script Inference. *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Long Papers*, [online] 1, pp.279–289. Available at: <https://arxiv.org/abs/1604.02993v2> [Accessed 10 Feb. 2023].

Purdy, C., Wang, X., He, L. and Riedl, M., (2018) Predicting Generated Story Quality with Quantitative Measures. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment* (Vol. 14, No. 1, pp. 95-101). [online] Available at: <https://dl.acm.org/doi/10.5555/3505378.3505392> [Accessed 13 Feb. 2023].

Qiu, X.P., Sun, T.X., Xu, Y.G., Shao, Y.F., Dai, N. and Huang, X.J., (2020) Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, [online] 6310, pp.1872–1897. Available at: <https://link.springer.com/article/10.1007/s11431-020-1647-3> [Accessed 10 Feb. 2023].

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D. and Sutskever, I., (2019) Language Models are Unsupervised Multitask Learners. OpenAI blog [online] , 1(8), p.9. Available at: <https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf> [Accessed 10 Feb. 2023].

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W. and Liu, P.J., (2020) Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, [online] 21, pp.1–67. Available at: <https://dl.acm.org/doi/10.5555/3455716.3455856> [Accessed 10 Feb. 2023].

Rashkin, H., Smith, E.M., Li, M. and Boureau, Y.L., (2018) Towards Empathetic Open-domain Conversation Models: a New Benchmark and Dataset. *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, [online] pp.5370–5381. Available at: <https://arxiv.org/abs/1811.00207v5> [Accessed 23 Oct. 2022].

Ratner, A.J., Ehrenberg, H.R., Hussain, Z., Dunnmon, J. and Ré, C., (2017) Learning to Compose Domain-Specific Transformations for Data Augmentation.

# **APPENDIX A**

# **RESEARCH PROPOSAL**

GENERATING A NARRATIVE WITH EXCELLENT STORY TELLING CAPABILITIES BY GIVING PROMPTS TO PRE-TRAINED MODELS

MOHAMMED AAMER

Research Proposal

May 2024

Abstract

With advent of advanced language models (PLM), significant advances have been made in text design problems. However, these models can only reflect a certain width of the output. PLM cannot create long stories because it does not understand narrative structure. Recent research on story generation has used the concept of clarity, which can create stories with multiple reasons for events. However, the scarcity of training data makes the optimization of PLM difficult. Even with appropriate treatment, definitive control is difficult to achieve. Therefore, it is not easy to create models that can create long stories. A recent study answers this question. This article shows a way to use rapid learning to create stories while maintaining quality control.

# 1. Introduction

The quality of machine authoring has increased in recent decades with the invention of large-scale pretraining models. Today they can produce texts that are very similar to human handwriting.

However, while large-scale PLM has demonstrated great potential in creating consistent and understandable documents, managing the creation remains a challenge (Keskar et al., 2019; Radford et al., 2019). A more comprehensive review of the published literature revealed issues such as conflicting topics and conflicting personalities (Fan et al., 2019; Bisk et al). These deficits are especially important for open reading activities that require collaboration, such as narrative construction.

Narratives created using these models have been shown to have absence of dialogue and. Cognitive skills. Although single sentences in the text may seem meaningful and flowing, when put together the whole narrative is not very cohesive (see et al., 2019; Goldfarb-Tarrant et al., 2020). Repetition of sentences in long texts will reduce the quality of the story (Yao et al., 2019). Preparation of content in different formats. Use the instructions. Use keywords and keywords using semantic frames. To get the most out of these projects, PLM often needs to fine-tune the data. One of the challenges in optimizing PLM is that in addition to the required information, the model also tends to examine events arising from the planned content and provide useful context from them (Fan et al., 2019). This causes a lack of diversity in the narratives created. In this context, special warnings are used to solve text problems. Study indicates that using guidelines, PLM can resolve current or next-generation tasks eliminating the need for optimization. But still some problems exist. Cues are specific to a task and difficult to adapt or reuse to new tasks. Even within the task, instructions may not be applicable to all situations in large groups.

# 2. Existing Research

The stories of the affected generation are examined from various perspectives. Researchers use ideas, genre, style, content, etc. to control story formation. They focused on the use of general concepts. Some studies try to use stories, narratives and plans for detailed management (Peng et al., 2018). This study was evaluated using the Five Element Stories dataset ROC Stories (Mostafazadeh et al., 2016). Later, some studies have experimented using long texts to control story formation (Fang et al., 2021).

Researchers also attempted to use QC control to guide the novella. (Fang et al., 2021) suggested creating stories based on the content of the story/sentence. (Rashkin et al., 2020) proposed a similar system with a special system and memory. (Sun et al., 2020) Create a storytelling by creating a summary for each part of the story. Each element is then expanded to form a complete story.

Most research in this area is on the basis of fine-tuning selected or constructed TX-based pre-learned language models (PLMs). In particular, GPT-2 has received great recognition in this field due to its unique structure for illicit texts. Recently, the use of GPT3 (Brown et al., 2020) for text has increased following the release of its API (Dou et al., 2021; Shakeri et al., 2021).

It is difficult to fine-tune PLM when there is not enough data. To solve this problem, researchers tried the plug-and-play method to control the creation of stories without proper editing (2022 Sen et al., 2022).

Another learning method is hint based that does not require proper correction. Some projects use multigenerational teaching (Brown et al., 2020). Others tried to obtain discrete instructions and continuous instructions. Some even attempted to use the workplace to generate cues for goal work (Vu et al., 2021).

# 3. Questions

In this research, we will try address the below points:

1. Story creation process along with quality measures needs to be optimized in PLM. Can this technique be used with rapid learning to create stories with fewer shots without requiring fine-tuning?
2. We have been using Prompt-based learning in many ways to generate text. Is it possible to extende this to story-making?
3. The previous system only used GPT2 as the base model. Can text be loaded using a new version (or newer) of GPT3?

# 4. Objectives

We try to apply the capabilities of GPT3 for long story generation which can be controllable.

Objectives:

* Perform a thorough review of available literature for long narrative generation and shot generation techniques.
* Attempt to develop a method for generation of long and short stories using few prompts
* To assess the generated narratives with the help of existing metrics and compare with them.

# 5. Significance

Story creation is an active area of ​​research. Although the short story has been studied extensively, its longer history has been less researched. Although optimization has been used as a method in previous studies, there is no research on creating stories without optimization. This study also explores recent developments and some inflections in cognitive-based learning. Writers can benefit from this by getting assistance for new ideas or overcome the condition known as writer's block.

# 6. Scope of Study

Below is our scope for this study:

* Completion time is 18 weeks after submission of the proposal of research.
* Open-source tool and data will be used for this project
* Publicly available compute sources will be used for the project.

# 7. Research Methodology

The project focuses on generation of text and the capability to learn of PLM.

## 7.1 Dataset

We make the use of below datasets:

* **ROC Stories**: This document was introduced by (Mostafazadeh et al., 2016) and is approximately 98,000 5 sentences long, including story titles. This information is widely used in short-form generation projects.
* **Writing Prompts**: This collection was introduced by (Fan et al., 2018b) and is an introduction to story writing alongside stories written by approximately 300,000 groups. The story was captured from reddit. These are long, multi-sentence stories, therefore useful for complex tasks such as generation of long form stories.

## 7.2 Data Preparation

Plan should be an example of a sentence. While sending articles and sharing written information via ROC Stories, no information is prepared. For this reason, the process needs to be extracted from the story data set and then displayed on the link. This pair of descriptive sentences can be used in many conversations.

One of two forms can be taken by outline instances:

* **Summary**An example of the structure here is the summary of the sentence. This article has been expanded from the abstract. For content extraction, it is recommended to use TextRank (Mihalcea and Tarau, 2004) for extracting the words that give the most information from the sentence.
* **Key phrases / Keywords**  
  Examples of structure here are the main subject and the expressions in the sentence. Articles consist of these keywords/keywords. For the extraction process, it is recommended to use RAKE (Rose et al., 2010) to remove important words from sentence.

## 7.3 Algorithms

### 7.3.1 Pre-trained Language Models (PLMs)

The method of transfer learning is use by PLMs. It’s the method of utilizing exiting knowldge to new jobs. It always uses a lot of written content for supervision training.

Pretraining for self-tracking of multiple unlabeled objects has become the most popular learning transformation in deep learning. Pretraining differs from other methods in that it uses anonymous data for traiing itself and is useful for several tasks, from optimization to microlearning. These models are trained to self-monitor using a variety of inappropriate texts.

These models can be used in many natural language applications such as answering questions, creating text, classifying text. Beautiful designs. Some of the most popular language patterns are:

* GPT3 (Brown et al., 2020)
* BERT (Devlin al., 2018)

7.3.2 Few-Shot Learning (FSL)

Using a small sample size and experience, people can easily identify new categories in the data. This is also known as mets learning. Learning in Few Steps is form of meta-learning. This way, learners are taught many related tasks during the meta-study period, with the aim of making new (but related) tasks useful with some small examples during the meta-analysis. A good way to solve the problem of learning a pair is to learn for several tasks and then train the specific tasks. It’s a method that always monitors learning process, which requires a lot of recording information for the purpose of training.

### 7.3.3 Prompt-Learning

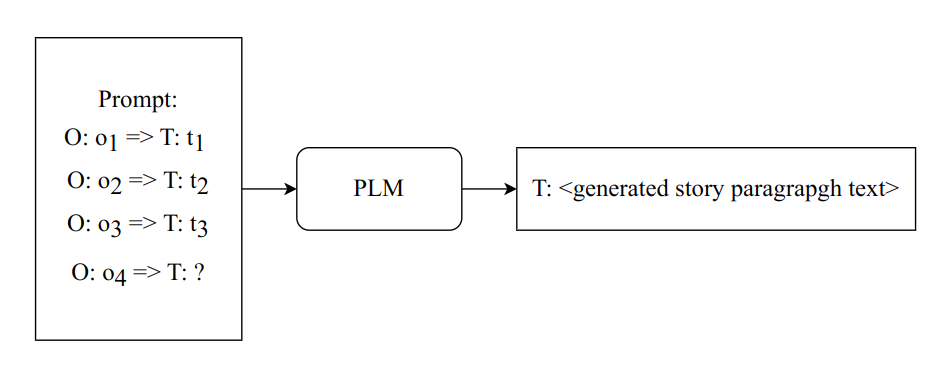
English Language Learning is a new course for teaching learning styles to ML models. When prompted, users say directly in the language of the task that they need previous language patterns to understand and complete. Whereas, Transformer always first performs the representation utilizing unsigned data and then uses the registered data to tune the model to use the necessary low-level operations. Warnings are user-written instructions that the model must follow. More inspiration may be needed depending on the complexity of the task being trained. Request engineering is the process of selecting appropriate requests or requests for required tasks. Sign-based learning has many advantages over pre-training and traditional methods. The main conclusion is that most of the lessons are done well with a small sample of labels.

## 7.4 Implementation

What we have implemented can be categorized into 2 steps:

1. Creating prompts for Few-Shot Learning – In this step, information is generated about a small set of sample pairs. Each example has a outline (o) and a paragraph (t).

Use examples of verbs with several conjugations to create incomplete sentences for new verbs; Verbs, examples with few conjugations, and sample questions are sent to the sample channel as emotional input. This formula returns the sentence generated by the given query.



**Fig. 7.4.1**

## 7.5 Assessment

Resulting stories will be assessed with various measurements. The study focuses solely on assessement using indicators which are automated. Book review of resulting stories is beyond this study.

The evaluation metrics are as follows:

* **ROUGE** (Lin, 2004) – Interpretation is similar to score of BLEU.
* **Perplexity (PPL)** (Fang et al., 2021; Jin et al., 2022) – Its used for computing complexity at word level.
* **Self-BLEU** (Zhu et al., 2018) – Used for measuring lexical diversity (intra-story)
* **BLEU** (Papineni et al., 2002) – Used for measuring n-gram overlap b/w the ground truth and generated texts.
* **DIST/distinct-n** (Li et al., 2015) - It quanitifis diversity in generation as a ratio of all generated n-grams to distinct n-grams.

Benchmarking is done against the following base:

* Outline-2-Story (**O2S**) (Fang et, 2021)
* Prompt Transfer for Text Generation (**PTG**) (Li et al., 2022)
* Summarize and Outlin and Elaborate (**SOE**) (Sun et al., 2020)

# 8. Requirements

## 8.1 Software

Below is the list of software that we will be utilizing for this work:

* Internet Explorer (Preferable Chrome)
* Integrated Development Environment (Preferable PyCharm)
* Python for coding
* NVIDIA - CUDA libraries
* DL Lib (TensorFlow, PyTorch, HuggingFace)

## 8.2 Hardware

We will be using the following hardware:

* PC with internet access and coding application
* GPUs to train the models

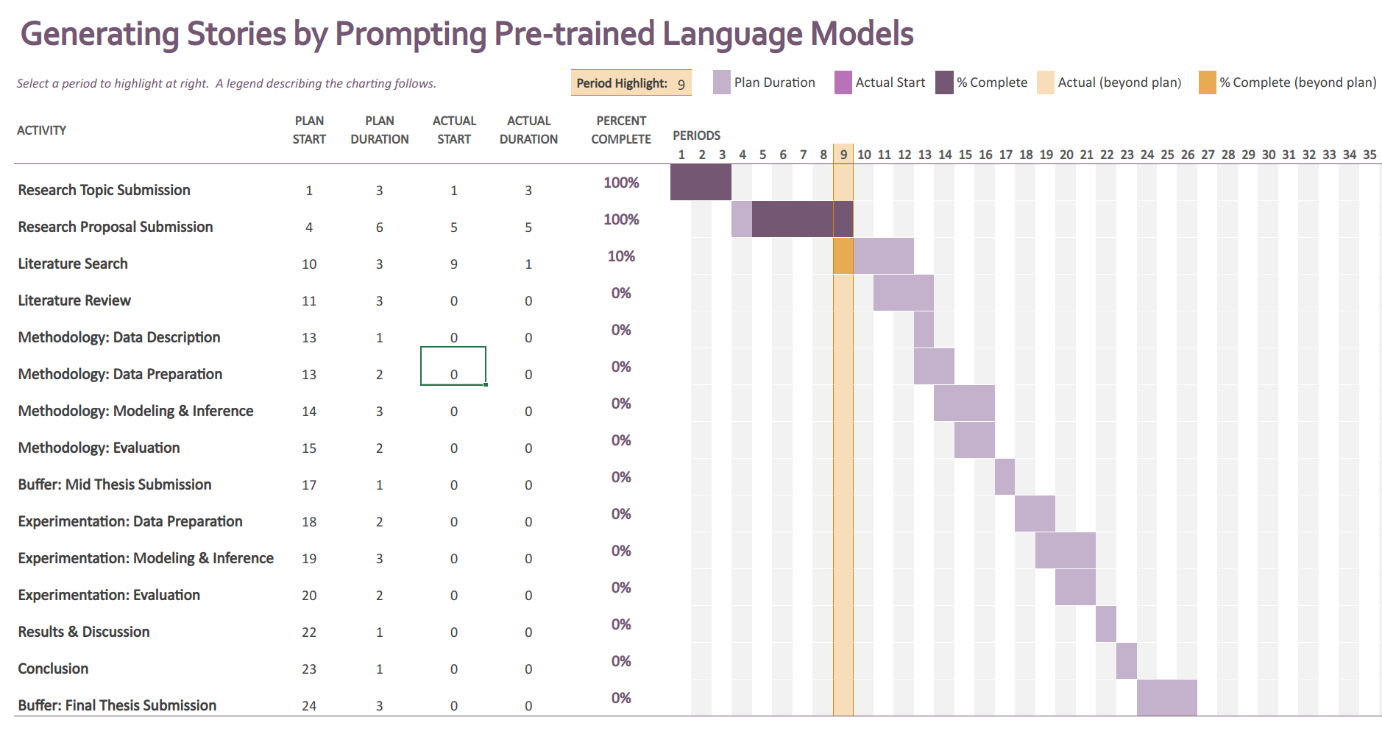
## 8.3 Dataset

We will be using the following datasets:

* ROC Stories dataset

# 9. Plan

## 9.1 Gantt Chart



**Fig. 9.1.1 (**1 Period = 7 days)

## 9.2 Risk & Exigency

Below is a list of risks and related situations in completing a thesis project.:

**Table 9.2.1**

|  |  |
| --- | --- |
| **Risk** | **Exigency** |
| Candidates cannot conduct research studies due to health problems or personal problems that may affect the duration. | Add buffer time & contact university for any additional support |
| Non-availability of local compute | Use cloud graphical processing units |

# References

Anon (2022) *ROCStories and the Story Cloze Test*. [online] Available at: <https://cs.rochester.edu/nlp/rocstories/> [Accessed 26 Oct. 2022].

Bakhtin, A., Deng, Y., Ott, M., Ranzato, M.’ A. and Szlam, A., (2021) Residual Energy-Based Models for Text. *Journal of Machine Learning Research*, [online] 22, pp.1–41. Available at: <http://jmlr.org/papers/v22/20-326.html>. [Accessed 23 Oct. 2022].

Bisk, Y., Holtzman, A., Thomason, J., Andreas, J., Bengio, Y., Chai, J., Lapata, M., Lazaridou, A., May, J., Nisnevich, A., Pinto, N. and Turian, J., (2020) Experience Grounds Language. *EMNLP 2020 - 2020 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, [online] pp.8718–8735. Available at: <https://arxiv.org/abs/2004.10151v3> [Accessed 23 Oct. 2022].

Bosselut, A., Celikyilmaz, A., He, X., Gao, J., Huang, P. sen and Choi, Y., (2018) Discourse-Aware Neural Rewards for Coherent Text Generation. *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, [online] 1, pp.173–184. Available at: <https://arxiv.org/abs/1805.03766v1> [Accessed 23 Oct. 2022].

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. *Advances in Neural Information Processing Systems*, [online] 33, pp.1877–1901. Available at: <https://commoncrawl.org/the-data/> [Accessed 23 Oct. 2022].

Chen, Z., Eavani, H., Chen, W., Liu, Y. and Wang, W.Y., (2019) Few-Shot NLG with Pre-Trained Language Model. [online] pp.183–190. Available at: <https://arxiv.org/abs/1904.09521v3> [Accessed 25 Oct. 2022].

Clark, E., August, T., Serrano, S., Haduong, N., Gururangan, S. and Smith, N.A., (2021) All That’s ‘Human’ Is Not Gold: Evaluating Human Evaluation of Generated Text. *ACL-IJCNLP 2021 - 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, [online] pp.7282–7296. Available at: <https://arxiv.org/abs/2107.00061v2> [Accessed 23 Oct. 2022].

Conneau, A. and Lample, G., (2019) Cross-lingual Language Model Pretraining. *Advances in Neural Information Processing Systems*, [online] 32. Available at: <https://arxiv.org/abs/1901.07291v1> [Accessed 25 Oct. 2022].

Dathathri, S., Madotto, A., Lan, J., Hung, J., Ai, U., Frank, E., Molino, P., Yosinski, J. and Liu, R., (2019) Plug and Play Language Models: A Simple Approach to Controlled Text Generation. [online] Available at: <https://arxiv.org/abs/1912.02164v4> [Accessed 25 Oct. 2022].

Devlin, J., Chang, M.W., Lee, K. and Toutanova, K., (2018) BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, [online] 1, pp.4171–4186. Available at: <https://arxiv.org/abs/1810.04805v2> [Accessed 23 Oct. 2022].

Dong, L., Yang, N., Wang, W., Wei, F., Liu, X., Wang, Y., Gao, J., Zhou, M. and Hon, H.-W., (2019) Unified Language Model Pre-training for Natural Language Understanding and Generation. *Advances in Neural Information Processing Systems*, [online] 32. Available at: <https://github.com/microsoft/unilm>. [Accessed 25 Oct. 2022].

Dou, Y., Forbes, M., Koncel-Kedziorski, R., Smith, N.A. and Choi, Y., (2021) Is GPT-3 Text Indistinguishable from Human Text? Scarecrow: A Framework for Scrutinizing Machine Text. [online] pp.7250–7274. Available at: <https://arxiv.org/abs/2107.01294v3> [Accessed 23 Oct. 2022].

Dziri, N., Madotto, A., Zaiane, O. and Bose, A.J., (2021) Neural Path Hunter: Reducing Hallucination in Dialogue Systems via Path Grounding. *EMNLP 2021 - 2021 Conference on Empirical Methods in Natural Language Processing, Proceedings*, [online] pp.2197–2214. Available at: <https://arxiv.org/abs/2104.08455v2> [Accessed 23 Oct. 2022].

Fan, A., Lewis, M. and Dauphin, Y., (2018a) Hierarchical Neural Story Generation. *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, [online] 1, pp.889–898. Available at: <https://arxiv.org/abs/1805.04833v1> [Accessed 23 Oct. 2022].

Fan, A., Lewis, M. and Dauphin, Y., (2018b) Hierarchical Neural Story Generation. *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, [online] 1, pp.889–898. Available at: <https://aclanthology.org/P18-1082> [Accessed 13 Sep. 2022].

Fan, A., Lewis, M. and Dauphin, Y., (2019) Strategies for Structuring Story Generation. *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, [online] pp.2650–2660. Available at: <https://arxiv.org/abs/1902.01109v2> [Accessed 23 Oct. 2022].

Goldfarb-Tarrant, S., Chakrabarty, T., Weischedel, R. and Peng, N., (2020) Content Planning for Neural Story Generation with Aristotelian Rescoring. *EMNLP 2020 - 2020 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, [online] pp.4319–4338. Available at: <https://arxiv.org/abs/2009.09870v2> [Accessed 23 Oct. 2022].

Guan, J., Huang, F., Zhao, Z., Zhu, X. and Huang, M., (2020) A knowledge-enhanced pretraining model for commonsense story generation. *Transactions of the Association for Computational Linguistics*, [online] 8, pp.93–108. Available at: <https://direct.mit.edu/tacl/article/doi/10.1162/tacl_a_00302/43540/A-Knowledge-Enhanced-Pretraining-Model-for> [Accessed 23 Oct. 2022].

Hu, Z., Yang, Z., Liang, X., Salakhutdinov, R. and Xing, E.P., (2017) *Toward Controlled Generation of Text*. Available at: <https://proceedings.mlr.press/v70/hu17e.html> [Accessed 25 Oct. 2022].

Hua, X. and Wang, L., (2020) PAIR: Planning and Iterative Refinement in Pre-trained Transformers for Long Text Generation. *EMNLP 2020 - 2020 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, [online] pp.781–793. Available at: <https://arxiv.org/abs/2010.02301v1> [Accessed 23 Oct. 2022].

Ji, H. and Huang, M., (2021) DiscoDVT: Generating Long Text with Discourse-Aware Discrete Variational Transformer. *EMNLP 2021 - 2021 Conference on Empirical Methods in Natural Language Processing, Proceedings*, [online] pp.4208–4224. Available at: <https://arxiv.org/abs/2110.05999v1> [Accessed 23 Oct. 2022].

Ji, H., Ke, P., Huang, S., Wei, F., Zhu, X. and Huang, M., (2020) Language Generation with Multi-Hop Reasoning on Commonsense Knowledge Graph. *EMNLP 2020 - 2020 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, [online] pp.725–736. Available at: <https://arxiv.org/abs/2009.11692v1> [Accessed 23 Oct. 2022].

Jin, Y., Kadam, V. and Wanvarie, D., (2022) Plot Writing From Pre-Trained Language Models. *INLG 2022*. [online] Available at: <https://arxiv.org/abs/2206.03021> [Accessed 24 Aug. 2022].

Keskar, N.S., Mccann, B., Varshney, L.R., Xiong, C., Socher, R. and Research, S., (2019) CTRL: A Conditional Transformer Language Model for Controllable Generation. [online] Available at: <https://arxiv.org/abs/1909.05858v2> [Accessed 23 Oct. 2022].

Li, J., Galley, M., Brockett, C., Gao, J. and Dolan, B., (2015) A Diversity-Promoting Objective Function for Neural Conversation Models. *2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2016 - Proceedings of the Conference*, [online] pp.110–119. Available at: <https://arxiv.org/abs/1510.03055v3> [Accessed 29 Oct. 2022].

Li, J., Tang, T., Nie, J.-Y., Wen, J.-R. and Zhao, W.X., (2022) Learning to Transfer Prompts for Text Generation. *NAACL 2022*, [online] pp.3506–3518. Available at: <https://arxiv.org/abs/2205.01543v2> [Accessed 11 Oct. 2022].

Li, J., Tang, T., Zhao, W.X., Wei, Z., Yuan, N.J. and Wen, J.R., (2021) Few-shot Knowledge Graph-to-Text Generation with Pretrained Language Models. *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, [online] pp.1558–1568. Available at: <https://arxiv.org/abs/2106.01623v1> [Accessed 25 Oct. 2022].

Li, X.L. and Liang, P., (2021) Prefix-Tuning: Optimizing Continuous Prompts for Generation. *ACL-IJCNLP 2021 - 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, [online] pp.4582–4597. Available at: <https://arxiv.org/abs/2101.00190v1> [Accessed 25 Oct. 2022].

Lin, C.-Y., (2004) *ROUGE: A Package for Automatic Evaluation of Summaries*. Available at: <https://aclanthology.org/W04-1013> [Accessed 29 Oct. 2022].

Lin, Z. and Riedl, M.O., (2021) Plug-and-Blend: A Framework for Controllable Story Generation with Blended Control Codes. *NAACL | NUSE | 2021*, [online] pp.62–71. Available at: <https://aclanthology.org/2021.nuse-1.7> [Accessed 31 Aug. 2022].

Liu, X., Zheng, Y., Du, Z., Ding, M., Qian, Y., Yang, Z. and Tang, J., (2021) GPT Understands, Too. [online] Available at: <https://arxiv.org/abs/2103.10385v1> [Accessed 25 Oct. 2022].

Mao, H.H., Majumder, B.P., McAuley, J. and Cottrell, G.W., (2019) Improving Neural Story Generation by Targeted Common Sense Grounding. *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, [online] pp.5988–5993. Available at: <https://aclanthology.org/D19-1615> [Accessed 31 Aug. 2022].

Mihalcea, R. and Tarau, P., (2004) TextRank: Bringing Order into Texts.

Mori, Y., Yamane, H., Shimizu, R. and Harada, T., (2022) Plug-and-Play Controller for Story Completion: A Pilot Study toward Emotion-aware Story Writing Assistance. [online] pp.46–57. Available at: <https://aclanthology.org/2022.in2writing-1.6> [Accessed 1 Oct. 2022].

Mostafazadeh, N., Chambers, N., He, X., Parikh, D., Batra, D., Vanderwende, L., Kohli, P. and Allen, J., (2016) A Corpus and Cloze Evaluation for Deeper Understanding of Commonsense Stories. *2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2016 - Proceedings of the Conference*, [online] pp.839–849. Available at: <https://aclanthology.org/N16-1098> [Accessed 13 Sep. 2022].

Papineni, K., Roukos, S., Ward, T. and Zhu, W.-J., (2002) Bleu: a Method for Automatic Evaluation of Machine Translation. *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL ’02*, [online] pp.311–318. Available at: <https://aclanthology.org/P02-1040> [Accessed 29 Oct. 2022].

Pascual, D., Egressy, B., Bolli, F. and Wattenhofer, R., (2020) Directed Beam Search: Plug-and-Play Lexically Constrained Language Generation. [online] Available at: <http://arxiv.org/abs/2012.15416> [Accessed 31 Aug. 2022].

Pascual, D., Egressy, B., Meister, C., Cotterell, R. and Wattenhofer, R., (2021) A Plug-and-Play Method for Controlled Text Generation. *Findings of the Association for Computational Linguistics, Findings of ACL: EMNLP 2021*, [online] pp.3973–3997. Available at: <https://aclanthology.org/2021.findings-emnlp.334> [Accessed 31 Aug. 2022].

Peng, N., Ghazvininejad, M., May, J. and Knight, K., (2018) Towards Controllable Story Generation. [online] pp.43–49. Available at: <https://aclanthology.org/W18-1505> [Accessed 25 Oct. 2022].

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D. and Sutskever, I., (2019) Language Models are Unsupervised Multitask Learners. [online] Available at: <https://github.com/codelucas/newspaper> [Accessed 23 Oct. 2022].

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W. and Liu, P.J., (2020) Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, [online] 21, pp.1–67. Available at: <http://jmlr.org/papers/v21/20-074.html>. [Accessed 25 Oct. 2022].

Rashkin, H., Celikyilmaz, A., Choi, Y. and Gao, J., (2020) PlotMachines: Outline-Conditioned Generation with Dynamic Plot State Tracking. *EMNLP 2020 - 2020 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, [online] pp.4274–4295. Available at: <https://aclanthology.org/2020.emnlp-main.349> [Accessed 31 Aug. 2022].

Rashkin, H., Smith, E.M., Li, M. and Boureau, Y.L., (2018) Towards Empathetic Open-domain Conversation Models: a New Benchmark and Dataset. *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, [online] pp.5370–5381. Available at: <https://arxiv.org/abs/1811.00207v5> [Accessed 23 Oct. 2022].

Rose, S., Engel, D., Cramer, N. and Cowley, W., (2010) Automatic Keyword Extraction from Individual Documents. *Text Mining: Applications and Theory*, [online] pp.1–20. Available at: <https://onlinelibrary.wiley.com/doi/full/10.1002/9780470689646.ch1> [Accessed 29 Oct. 2022].

le Scao, T. and Rush, A.M., (2021) How Many Data Points is a Prompt Worth? *NAACL-HLT 2021 - 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference*, [online] pp.2627–2636. Available at: <https://arxiv.org/abs/2103.08493v2> [Accessed 25 Oct. 2022].

See, A., Pappu, A., Saxena, R., Yerukola, A. and Manning, C.D., (2019) Do Massively Pretrained Language Models Make Better Storytellers? *CoNLL 2019 - 23rd Conference on Computational Natural Language Learning, Proceedings of the Conference*, [online] pp.843–861. Available at: <https://arxiv.org/abs/1909.10705v1> [Accessed 23 Oct. 2022].

Shakeri, H., Neustaedter, C. and DiPaola, S., (2021) SAGA: Collaborative Storytelling with GPT-3; SAGA: Collaborative Storytelling with GPT-3. *Companion Publication of the 2021 Conference on Computer Supported Cooperative Work and Social Computing*. [online] Available at: <https://doi.org/10.1145/3462204.3481771> [Accessed 1 Oct. 2022].

Shen, T., Lei, T., Barzilay, R., Jaakkola, T. and Csail, M., (2017) Style Transfer from Non-Parallel Text by Cross-Alignment. *Advances in Neural Information Processing Systems*, [online] 30. Available at: <https://github.com/shentianxiao/language-style-transfer>. [Accessed 25 Oct. 2022].

Shin, T., Razeghi, Y., Logan, R.L., Wallace, E. and Singh, S., (2020) AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. *EMNLP 2020 - 2020 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, [online] pp.4222–4235. Available at: <https://arxiv.org/abs/2010.15980v2> [Accessed 25 Oct. 2022].

Song, K., Tan, X., Qin, T., Lu, J. and Liu, T.Y., (2019) MASS: Masked Sequence to Sequence Pre-training for Language Generation. *36th International Conference on Machine Learning, ICML 2019*, [online] 2019-June, pp.10384–10394. Available at: <https://arxiv.org/abs/1905.02450v5> [Accessed 25 Oct. 2022].

Su, Y., Wang, X., Qin, Y., Chan, C.-M., Lin, Y., Wang, H., Wen, K., Liu, Z., Li, P., Li, J., Hou, L., Sun, M. and Zhou, J., (2021) On Transferability of Prompt Tuning for Natural Language Processing. *NAACL 2022 - 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference*, [online] pp.3949–3969. Available at: <http://arxiv.org/abs/2111.06719> [Accessed 25 Oct. 2022].

Sun, X., Fan, C., Sun, Z., Meng, Y., Wu, F. and Li, J., (2020) Summarize, Outline, and Elaborate: Long-Text Generation via Hierarchical Supervision from Extractive Summaries. [online] Available at: <https://arxiv.org/abs/2010.07074v2> [Accessed 23 Oct. 2022].

Tan, B., Yang, Z., Al-Shedivat, M., Xing, E.P. and Hu, Z., (2020) Progressive Generation of Long Text with Pretrained Language Models. *NAACL-HLT 2021 - 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference*, [online] pp.4313–4324. Available at: <https://arxiv.org/abs/2006.15720v2> [Accessed 23 Oct. 2022].

Vaswani, A., Brain, G., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., (2017) Attention is All you Need. *Advances in Neural Information Processing Systems*, 30.

Vu, T., Lester, B., Constant, N., Al-Rfou, R., Cer, D. and Research, G., (2021) SPoT: Better Frozen Model Adaptation through Soft Prompt Transfer. [online] pp.5039–5059. Available at: <https://arxiv.org/abs/2110.07904v2> [Accessed 25 Oct. 2022].

Xu, J., Ren, X., Zhang, Y., Zeng, Q., Cai, X. and Sun, X., (2018) A Skeleton-Based Model for Promoting Coherence Among Sentences in Narrative Story Generation. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018*, [online] pp.4306–4315. Available at: <https://arxiv.org/abs/1808.06945v2> [Accessed 23 Oct. 2022].

Xu, P., Patwary, M., Shoeybi, M., Puri, R., Fung, P., Anandkumar, A. and Catanzaro, B., (2020) MEGATRON-CNTRL: Controllable Story Generation with External Knowledge Using Large-Scale Language Models. *EMNLP 2020 - 2020 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, [online] pp.2831–2845. Available at: <https://arxiv.org/abs/2010.00840v1> [Accessed 23 Oct. 2022].

Yao, L., Peng, N., Weischedel, R., Knight, K., Zhao, D. and Yan, R., (2019) Plan-and-Write: Towards Better Automatic Storytelling. *Proceedings of the AAAI Conference on Artificial Intelligence*, [online] 3301, pp.7378–7385. Available at: <https://ojs.aaai.org/index.php/AAAI/article/view/4726> [Accessed 23 Oct. 2022].

Zellers, R., Holtzman, A., Rashkin, H., Bisk, Y., Farhadi, A., Roesner, F., Choi, Y. and Allen, P.G., (2019) Defending Against Neural Fake News. *Advances in Neural Information Processing Systems*, [online] 32. Available at: <https://rowanzellers.com/grover> [Accessed 23 Oct. 2022].

Zhang, Y., Sun, S., Galley, M., Chen, Y.-C., Brockett, C., Gao, X., Gao, J., Liu, J. and Dolan, B., (2019) DialoGPT: Large-Scale Generative Pre-training for Conversational Response Generation. [online] pp.270–278. Available at: <https://arxiv.org/abs/1911.00536v3> [Accessed 23 Oct. 2022].

Zhao, J.J., Kim, Y., Zhang, K., Rush, A.M. and Lecun, Y., (2018) Adversarially Regularized Autoencoders. [online] pp.5902–5911. Available at: <https://proceedings.mlr.press/v80/zhao18b.html> [Accessed 25 Oct. 2022].

Zhu, Y., Lu, S., Zheng, L., Guo, J., Zhang, W., Wang, J. and Yu, Y., (2018) Texygen: A benchmarking platform for text generation models. *41st International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2018*, pp.1097–1100.

Ziegler, Z.M., Melas-Kyriazi, L., Gehrmann, S. and Rush, A.M., (2019) Encoder-Agnostic Adaptation for Conditional Language Generation. [online] Available at: <https://arxiv.org/abs/1908.06938v2> [Accessed 25 Oct. 2022].

Zou, X., Yin, D., Zhong, Q., Yang, H., Yang, Z. and Tang, J., (2021) Controllable Generation from Pre-trained Language Models via Inverse Prompting. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.2450–2460.