Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

Robin C. Staff

What a Twist

Using Deep Neural Networks to Generate Plot Twists

Master's thesis in Computer Science
Supervisor: Björn Gambäck
July 2021

NTNU
Norwegian University of
Science and Technology

Robin C. Staff

What a Twist

Using Deep Neural Networks to Generate Plot Twists

Master's thesis in Computer Science
Supervisor: Björn Gambäck
July 2021

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

Abstract

Storytelling is a fundamental part of being human. Over thousands of years, stories have been told and passed down. It is one of the things that set us apart from animals. However, can machines be made to do this? The answer is yes. Over the last century, many different techniques were tested, leading to some exciting results. This Master's Thesis takes a hierarchical neural network system and aims to modify it to generate a compelling plot twist. This project also fine-tunes a pre-trained transformer for the same purpose of generating a twist.

Plot twists are unique because they require an understanding of a premise and a subversion of audience expectation. This is linguistically advanced and not something machines are widely known to be able to do. In this project, existing datasets for story generation training were considered. Since they were not found to be suitable, two datasets of twists were created from a variety of sources. The challenge then became the low amount of twists compared to how many entries are needed to train a neural network.

Since compiling enough examples of twists to train a neural network was impossible, fine-tuning a pre-trained transformer was a much more feasible approach. By fine-tuning GPT-2 on a dataset of twists compiled for this project, twist prompts were generated, which were able to make the human judges doubt whether or not they were generated.

Sammendrag

Det å skape fortellinger er en grunnleggende del av det å være et menneske. I mange tusen år har fortellinger blir delt fra person til person. Men kan datamaskiner nå bidra til dette? I de siste hundre år har mange teknikker for å automatisk generere fortellinger eller handlinger blitt forsøkt. Dette har ført til noen spennende resultater. Denne masteroppgaven tar et hierarkisk nevralt nettverk og forsøker få det til å generere en god handlingsvri (engelsk: plot twist). Prosjektet prøver også å finjustere en pre-trent transformator til formålet å generere en handlingsvri.

Handlingsvrier er spesielle innenfor forfatterskap fordi man må forstå premisset for handlingen, og publikum sine forventninger, for å kunne skrive en god handlingsvri. Det krever også god språkforståelse, og er totalt sett ikke noe datamaskiner er kjent for å kunne gjøre enda. I denne masteroppgaven ble eksiterende datasett som var laget for handlingsgenerering vurdert til formålet å trene nevrale nett som skulle generere handlingsvrier. Ingen av de eksisterende var brukbare til dette, og dermed ble to datasett bygget til dette prosjektet ved å bruke flere kilder. Utfordringen deretter, ble at det ble klart at det var umulig å finne mange nok eksisterende eksempler på handlingsvrier til å effektivt trene et nevralt nettverk.

Fordi det ble umulig å finne nok handlingsvrier til å trene et nevralt nettverk, ble å finjustere en pre-trent transformator en bedre løsning. GPT-2 ble finjustert på et datasett av handlingsvrier bygget for dette prosjektet. Handlingsvriene som så ble generert var gode nok til å så tvil blant mennesker i en blindtest som ble utført.

Acknowledgment

This Master's Thesis is the final part of my Master's Degree in Computer Science at NTNU Trondheim, spring 2021. The thesis concerns Story Generation and researching how well a machine can generate a twist that appeals to humans.

My supervisor has been Björn Gambäck.

I would like to thank my supervisor for always being there when I needed advice and setting meetings even on his days off work, and always having good examples and relevant literature when I was stuck on a problem. He even offered to meet me on midsommarafton (this was thankfully not needed), which was very generous as he is Swedish.

I want to thank my fellow student and brother Alexander Staff for lending an ear and advice when I was stuck on a Machine Learning problem. And I would like to thank my parents, who encouraged me through this challenging semester.

The computations for this project were performed on resources provided by the NTNU IDUN/EPIC computing cluster. [1]

Contents

Figures

Tables

Acronyms

GPT-2	Generative Pre-trained Transformer 2. iii, v, xiii, 2, 5, 6, 12, 13, 31, 32, 35, 42, 44, 46–55, 59–63, 66–68
GPT-3	Generative Pre-trained Transformer 3. 13, 60, 66, 68
LSTM	Long short-term memory. 10, 11
NLP	Natural Language Processing. 7, 13, 27
PAT	Planning Advice Themes. 19
RNN	Recurrent neural network. 9–11, 14
TRAM	Transform Recall Adapt Method. 19, 20

Chapter 1

Introduction

This project was initially inspired by experimental results from Pérez y Pérez and Sharples [2]. In one of the generated stories of the story generator MEXICA, there is a plot twist of sorts: the main characters, a princess and a knight, fall in love. However, when she sees the knight's tattoo, the princess realises that the knight she has fallen for is part of the fraternity that killed her father. Would it be possible to create a plot twist generator that was more specialised towards generating twists? Over the years, different approaches have been taken towards generating stories. Some focus on understanding and recreating a human process of writing, while others try to abstract aspects of stories so that they can more easily be programmed.

(This paragraph contains spoilers for The Sixth Sense, Harry Potter and the Goblet of Fire, and Frozen.) The study of story generation can be motivated by better understanding the mechanics of a story and story creation by humans as in Dehn [3]. Sometimes a part of it is trying to analyse how humans receive stories as in Wilmot and Keller [4]. A *plot twist* is a complex concept that involves all of these. Some storytellers are famed for their plot twists. For instance, M. Night Shyamalan's (The Sixth Sense, Signs, The Village, The Last Airbender) early career was made up of movies where the twist was the main feature. In The Sixth Sense, the idea for the twist was so good that they had to work extra hard to ensure the rest of the film could be considered good even without it[1]. Everything served to build up to the reveal. In other stories, the twist is a more subtle part of the story. In Harry Potter and the Goblet of Fire, the fact that Mad-Eye Moody turns out to be Barty Crouch Jr. in disguise is often forgotten, and people will refer to everything he did as done by "Mad-Eye Moody". In that case, the premise that wizards can disguise themselves entirely as someone else is introduced two books earlier in the series. There is also the question of whether the audience buys the twist. In The Sixth Sense, people were stunned and delighted when it turned out Bruce Willis's character was one of the ghosts the main character could see all along. The premise is simple: the main character can see dead people. The twist made sense on a rewatch and was staring the audience in the face the whole time. In

[1]Interview with M. Night Shyamalan

the movie Frozen, people are split on if it makes sense on a rewatch that Prince Hans was evil the whole time. These are considerations when trying to figure out how to generate a good twist.

1.1 Background and Motivation

Story generators have a pretty long history at this point. In many cases, their effectiveness and usefulness have been measured in how human-like their output seems. Other criteria have been to judge if the output is considered excellent or compelling. Researchers have tried to make them better by better mimicking the writing process of a human. The idea for this project came out of wondering if story generators were now able to use a complex literary technique, the plot twist. When written by a human, the plot twist requires planning, an understanding of manipulating the audience, and an understanding of set-up and pay-off. For a twist to work as intended and surprise the audience, it has to be original somehow. Many of the most known twists were unique for their time.

All of this amounts to make it a tall order for story generators. At this stage, they are impressive if they can generate something compelling or at least coherent and a little interesting. However, Neural Networks are powerful and unpredictable. With a suitable dataset, there is no telling what can be coaxed out of running experiments. Finding or creating a suitable dataset is the first big hurdle for this project. Is it possible to create a good dataset for generating twists? Are there even that many twists in existence? How many twists can you remember having experienced?

In the pre-study (Staff [5]) for this project, no example could be found of someone who had tested this. Nevertheless, there were some exciting leads to examine. The Hierarchical Neural Network of Fan *et al.* [6] was very promising as it was writing-prompt based, and this could be tested with a dataset of twisty prompts. Other systems have also been considered in this project, and some were tested, such as GPT-2. The ultimate aim is to generate interesting and original twists and test them on human judges against human written twists in a blind test.

1.2 Storytelling and Twists

The basic parts of storytelling consists of a plot, characters, setting, conflict, and narrative point of view. Structuralist narratology, as defined by Todorov [7] says a story's meaning develops from its overall structure. Todorov coined the term *narratology* to mean *the science of narratives*. When analysing a story, he split it into two levels of descriptions, the *story* and the *discourse*. Story consisted of the logic of action and syntax of characters, and discourse consisted of tenses, aspects, and narrative modes. The story is the set of events that happen in a given tale. The discourse is how these events are told: the order, emphasis, and pacing. Plot is understood to mean the set of story events linked by causality.

Modern narratologists believe narrative must be seen from a more cognitive perspective than just structure. Herman [8] proposes looking at narrative not as a story but as a way of reasoning about reality according to story logic. In computational terms, there seemingly is no consensus about what constitutes a narrative. Different projects have tried wildly different approaches with a focus on different aspects of stories. Some focus more on planning and premise, while others focus more on excitement or characters. When analysing human-written texts or story generators, a few things need to be considered, such as the level of granularity that abstracts the universe. Three levels of granularity are defined.

1. The universe of all entities and events.
2. The subset of that universe that is structured by time and causality.
3. The rendering of the narrative.

1.2.1 What is a Plot Twist

A plot twist is a writing trope where the audience experiences a radical change in the expected outcome. A twist often comes at the end of a story and changes how the audience sees the story's events in hindsight. To ensure the audience accepts the twist, the writer has to make the story work in the context of the twist, so it does not seem like cheating. Foreshadowing is a tool to prepare the audience for the reveal. To make the audience buy the initial understanding of events, the writer might withhold information or give them misleading information as detailed in Singleton *et al.* [9].

Many people do not want to know a twist before experiencing a book or movie because the effect of the twist will be gone, and they will not be surprised. Even so, one study has shown that learning them beforehand does not ruin the experience according to Lehrer [10]. Describing a twist in a story is often called a "spoiler", as it is thought to spoil the experience. In this project, where necessary, there might be a "spoiler warning" so the reader can avoid reading real examples of twists.

There are a few common ways in which plot twists are often used in stories. Knowing them can be helpful to analyse ways in which their mechanics might be abstracted for generation.

Discovery

Discovery is when the protagonist has a shocking revelation about their own or another character's nature or identity. This category encompasses some of the most famous twists in cinema. (*Spoiler Warning* for Star Wars, The Sixth Sense and Fight Club) In the second Star Wars film, Luke Skywalker discovers that he is the son of the villain, Darth Vader. In The Sixth Sense, the main character, a child psychologist, struggles with not having been able to help a former patient who shot him at the film's start. Throughout the film, he redeems himself by helping a child. This child believes he is seeing ghosts of dead people walking about, not knowing they are dead. In the end, the main character finds out that he died in the shooting at the beginning and that he is one of the ghosts only the boy he is trying to help can see. In Fight Club, the main character learns that the man who is helping him (or misleading him depending on how you read the work) is his own multiple personality.

Flashback

Flashbacks are commonly used to show an important event but adding information about it that the audience previously did not have. The key is that the new information fits with what the audience already knows but casts it all in a new light. This is common in murder mysteries, where the audience was given selective information the first time around. Then when the detective is explaining at the end, a flashback is shown of the actual events.

Unreliable Narrator

In most stories, the protagonist is seen as a reliable narrator, or their view of events is at least never challenged. This can be used where the writer shows things from their subjective perspective and, with the audience on the protagonist's side, reveals that the protagonist had been lying or is delusional.

Red Herring

A red herring is something in the story put there intentionally to mislead the audience and take their attention away from the things that would allow them to notice the twist. In crime fiction, a writer might put in one or more obvious "killers" so that the actual killer can move through the story unnoticed by the reader until the reveal.

False Protagonist

In some stories, the character presented as the main character for some, or even most, of the story is killed off or sidelined unexpectedly before the end. This is not the same as the protagonist dying at the very end when the story is over. Instead, they die before the story is over.

Non-linear or Reverse Narrative

In these stories, events in the story are told out of order or even just backwards. This enables the writer to keep some critical information hidden until the right moment by careful editing. In some stories, the reveal itself is that things were out of order.

1.3 Goals and Research Questions

Goal Investigate if story generation is at the point where a system can effectively generate plot twists.

A suitable dataset must be found or made to investigate this goal and then used in experiments with state-of-the-art systems. The state of the field has to be researched and an experimental plan made.

From the research motivation, three research questions were formulated for the project that will be explored in this thesis.

Research question 1 Are there suitable datasets for generating twists? If no, can one be made?

This question asks not only if the dataset exists but if the data itself even exists. Datasets for training systems in story generation usually are general so that they can be large. Twists are specific to a few stories and therefore must be sorted somehow.

Research question 2 Can a Hierarchical Neural Network be made to generate a twist?

For this research question Fan *et al.* [6] will be used as a starting point as it is a prompt-based system. Twists can be boiled down to prompts which makes it easier to distil the concept for training.

Research question 3 Can GPT-2 be made to generate a twist?

Because pre-trained transformers are the cutting edge in text generation, it will be interesting to compare how GPT-2 handles this problem to the specifically trained Fairseq system of Fan *et al.* [6].

1.4 Research Method

This project takes a two-pronged approach to its research methods. The first one consisted of a comprehensive literature review to get a solid overview of a fascinating and niche field of machine learning. This was necessary to build up a strong understanding of previous projects and state of the art. Also, within the field, suitable datasets are a big part of what makes specific experiments possible. This includes the experiments done for this project, which is what defined the second prong. The experiments were an attempt to add exciting results to the field.

1.5 Contributions

This project aims to contribute knowledge about the possibilities of state of the art story generation.

C1 Generated prompts featuring twists using GPT-2.

C2 A dataset of twists based on TV Tropes, IMDb and Wikipedia.

C3 An overview of how to use this project's Twist dataset to fine tune GPT-2 for generating twists.

C4 A comprehensive literature review of the existing datasets available within the field of story generation featuring twists.

C5 A ground up introduction to the field of story generation with regard to the challenges of good storytelling.

1.6 Thesis Structure

Chapter 2 Introduces the theory needed to understand the work in this project.

Chapter 3 Presents several projects in story generation that are of relevance to this project because they have solved important problems in the field.

Chapter 4 This chapter details datasets, data sources and the work done to compile data for this project.

Chapter 5 Gives a detailed overview of the architecture of the implemented systems.

Chapter 6 Details the experimental plan, and the results.

Chapter 7 Gives an evaluation of the experimental results.

Chapter 8 Gives a conclusion to the project and the possible future work that lies ahead in this field.

Chapter 2

Background Theory

This chapter looks at some relevant theories about stories and how different projects have attempted to generate them so far. The creation of story generators is a complicated task owing to how many features play a part in shaping a story, as seen in Leon *et al.* [11]. These features include emotions, characters, locations, intentions, narrative arc and character arc. Some of these need implicit values, and some need explicit values when generating a story. The range of acceptable values is high, so finding optimal values resulting in good stories is also challenging. Determining which values to use and quantify these features analyzing human-made stories is an excellent place to start.

2.1 Natural Language Processing

A computer can not understand raw text. It has to be represented in some way that makes it possible for computer architectures to process and interpret the text. Because a computer can do many operations in a short amount of time, a computer that can understand text contents is useful. In many cases, many texts have to be searched or condensed to solve some task, for instance, in law, where paralegals normally comb through an untold amount of documents during discovery. The field of research aiming to solve computer processing of text is called Natural Language Processing (NLP). A subfield of NLP is natural-language generation.

2.1.1 Feedforward Neural Network

A feedforward neural network is a neural network where connections between the nodes are not cyclical. This is the big difference to its descendant, recurrent neural networks. Feedforward neural networks are the simplest type of neural networks and, therefore, the easiest way to understand how they work. The information moves only one way, from input nodes, through the hidden nodes and to the output nodes. There is only a single output layer in a single-layer perceptron network, and the inputs are fed directly through these nodes via a series of weights. In each node, the sum of the weight and the input value is calculated, and if this sum is above the threshold, often 0, the neuron fires and takes the activated value, often 1. If the value is below the threshold, the node takes the inactive value, often -1. A perceptron is a network with just one node. Perceptrons can be trained by an algorithm called the *delta rule*.

The delta rule calculates the errors between calculated output and sample output data and uses this to adjust the weights. This is a form of gradient descent. *Gradient descent* is an optimization algorithm for finding the local minimum of a differentiable function. A single layer perceptron is only capable of learning linearly separable patterns. A linearly separable pattern can be understood by thinking of a plane with two different colour dots. So long as a line in the plane separates all dots of one colour from all the dots of the other color, the dots are considered linearly separable. See figure 2.1which shows an example of linearly separable data.

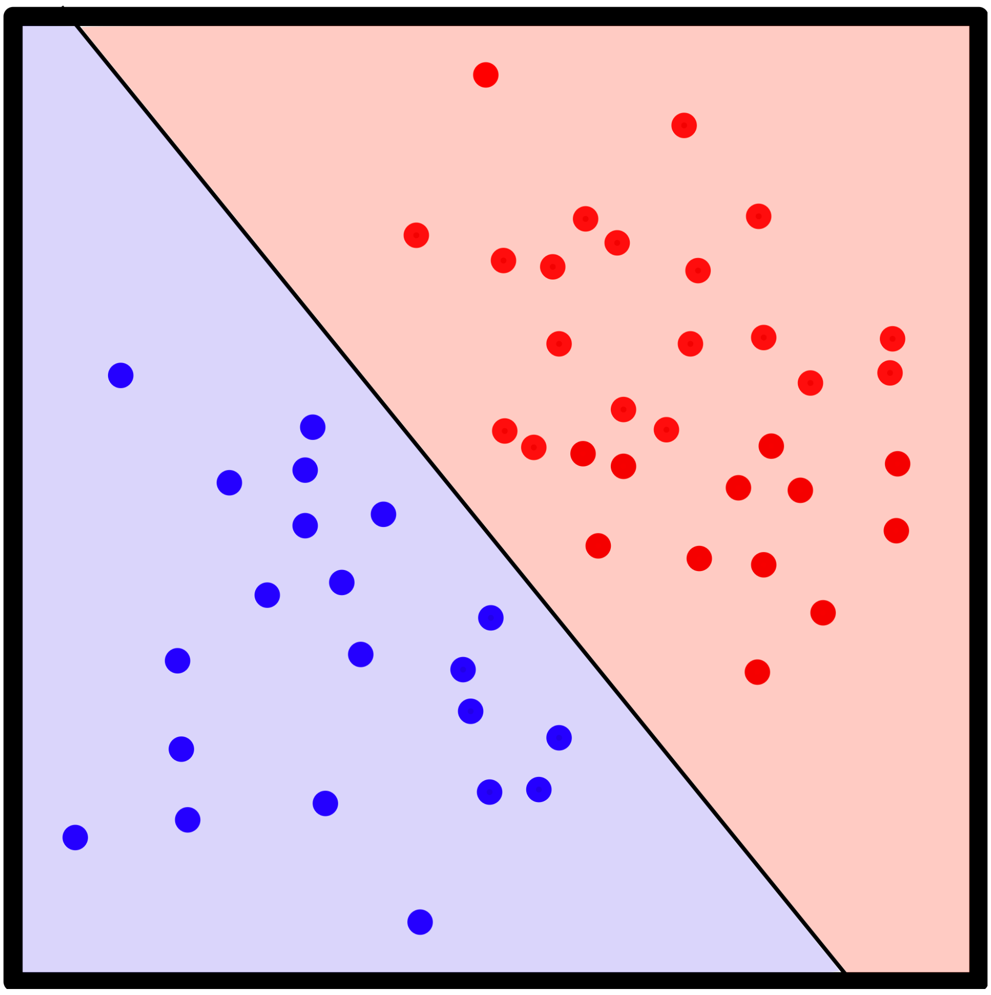


Figure 2.1: A line that separates all red and blue dots exists.[1]

A single layer neural network can compute a continuous output, such as a logistic function (equation 2.1). This makes the single-layer network the same as the *logistic regression model*. In statistics, this model is used to model the probability of certain events, for instance, win/lose or healthy/sick.

$$f(x) = \frac{1}{1 + e^x} \tag{2.1}$$

A multi-layer perceptron is made up of at least three layers of nodes: an input layer, an output layer and a hidden layer. Multi-layer perceptrons use a supervised learning technique called backpropagation. Intuitively, backpropagation is a process where during training, the output layer is checked for how far off each node is and then updates to the weights of each node propagate backwards from the output through the network. However, since the network is trained on more than one example, this has to be done for the average error across the training data. A neural network has to be big enough to encompass the differences in the training data. After a large number of training cycles, the network should, if possible given the training data, converge to a state where the error is low. For backpropagation to be possible, the network has to have a differentiable activation function. It calculates the derivative of the error function with respect to the network weights to figure out how to change the weights, so the error decreases. A possible weakness in this method is overfitting to training data (see figure 2.2 which shows an example of overfitting). This is where the model ends up trained to be very specific to training data and, therefore will end up less correct on different data than if it was more generally trained. In a statistical model, this is when it contains more parameters than can be justified by the data.

Other weaknesses are, the speed of convergence can be too low, making training time too long, and it is possible to end up in a local minimum of the error function, where the global minimum represents the optimal solution.

2.1.2 Natural Language Processing in Recurrent Neural Networks

The main problem with a feedforward neural network for language processing is that it has no memory. Without memory, when texts is split into words for processing, the network will only know the word it is currently processing. However, getting meaning from a text requires an understanding of the sequence of words. To solve this problem, an Recurrent neural network (RNN) can be used. RNNs are good at processing sequence data because they have a way of remembering previous input. In essence, this is a feedforward network with a loop that feeds previous output back into the system. This solution comes with some challenges related to a problem called the *vanishing gradient problem*. When training neural networks with gradient-based training and backpropagation, the gradient propagates backwards through the network. The error becomes much smaller each layer

[1]Source and licence for figure 2.1
[2]Source and licence for figure 2.2

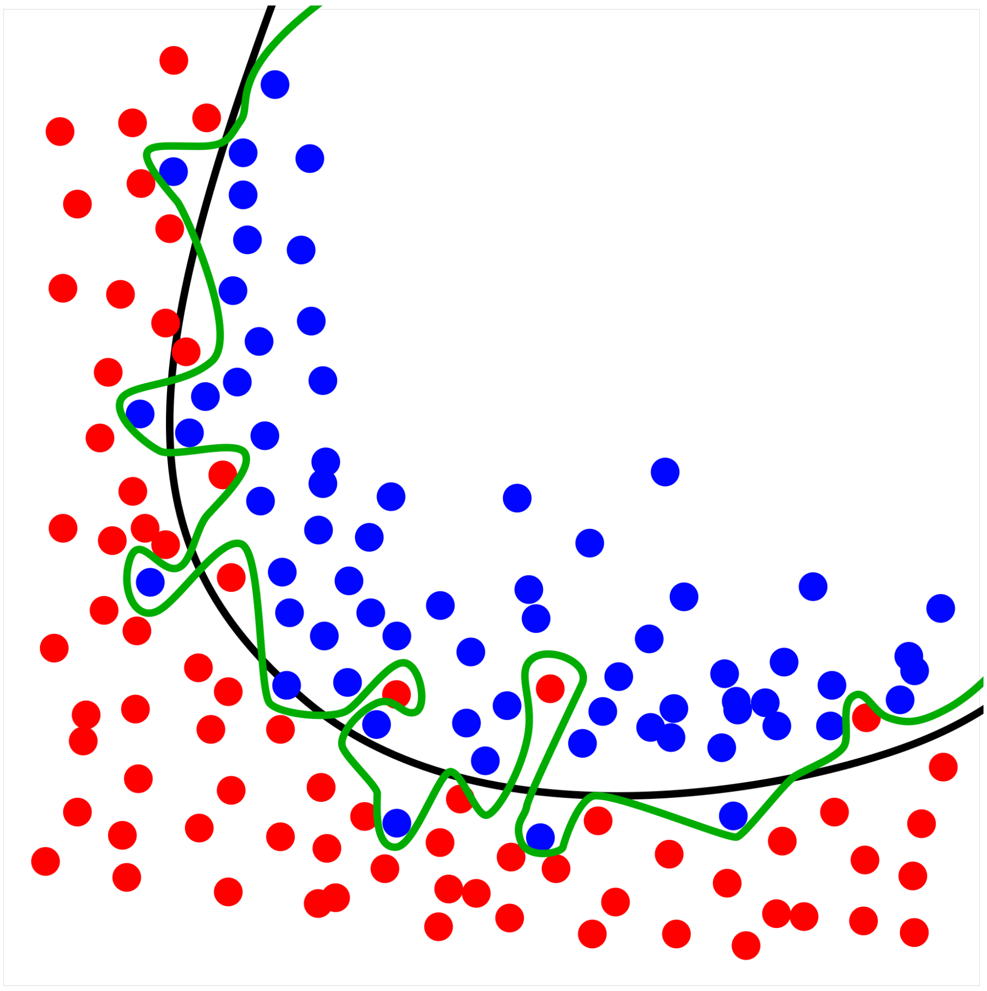


Figure 2.2: The green line is the overfitted model. The black line would be a more ideal general model.[2]

back until the early layers are not learning anything from each example. In the same way, as the output of the last part of the sequence is fed back, the earliest outputs become less and less relevant to the current state of the system for each loop. To mitigate this and give RNNs longer memory, solutions like Long short-term memory (LSTM) have been developed. In a standard RNN the back-propagated gradients can "vanish" if they trend to zero or "explode" if they trend to infinity because the computations in the process use finite precision numbers. With LSTM the vanishing problem is partially solved by allowing gradients also to flow unchanged. Unfortunately, LSTM does not solve the exploding gradient problem.

Figure 2.3 shows an LSTM cell. It consists of three gates and a cell state. The

[3]Source and licence for figure 2.3

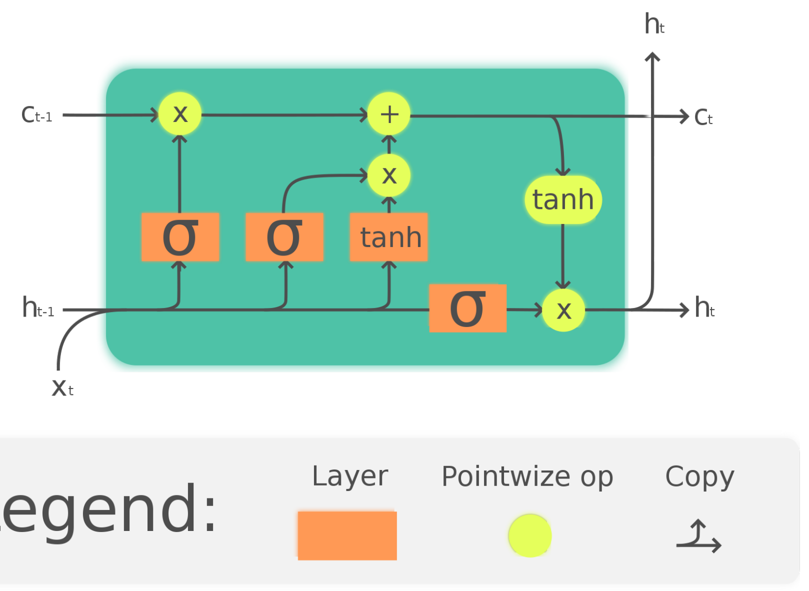


Figure 2.3: This special RNN architecture solves the vanishing gradient problem.[3]

cell state is the C_{t-1} to C_t channel running along the top of the figure. At each timestep, the next LSTM can choose to read from the cell, write to the cell or reset it using one of the three gates; the input gate, the forget gate, or the output gate.

2.2 Transformers

In the 2017 paper *Attention Is All You Need* by Vaswani *et al.* [12], researchers at Google detailed a new way of dealing with sequential data called the Transformer architecture. Previously, Long short-term memory (LSTM) had been used to deal with the vanishing gradient problem. Attention is used by making the encoded input of the network most relevant to the current step of output available to that step. The steps in the sequence are remembered and used in their corresponding output generation step. This is more complex than a standard RNN and takes even longer to train. The transformer architecture does away with the recurrent part of the neural network and only deals in vectors describing the position in the sentence and position in an embedding space. The embedding space can be seen as a map of clusters of words with similar meanings. This is used to map words to vectors. In addition to this, a positional encoding is added so that the system can catch the possibility of words changing their meaning based on their placement in a sentence.

The transformer is split into a series of encoder components and decoder components that run in parallel.

In the encoder, the input is then sent through a multi-headed attention layer. The attention layer tells the system which part of the system it should focus. For every word an attention vector is computed, which gives the contextual relationship of each word to the other words in the sentence. This attention vector is then sent through a feed-forward network that transforms it into something sent to the next encoder block.

In the decoder during training, the desired output is sent in as an embedding space vector with a positional vector added to it in the same way as in the encoder. It is then sent through a similar process as the input in the encoder, with an exception. There is an encoder to decoder attention block that looks at the attention vectors for each word of both the input and desired output to map them to each other. Every word in both the input and output is mapped to each other. There is also a masked attention block which ensures that when predicting the next word of the output, it cannot see the subsequent work of the output already. It only has access to the previous words.

The attention blocks are called multi-headed as they take an input of three matrices Q, K and V. Q meaning queries, K keys and V values. To compute what is called *Scaled Dot-Product Attention*, the dot product of the queries and keys of dimension d_k is computed. Each dot product is then divided by $\sqrt{d_k}$, and a softmax function is applied to obtain the weights on the values. The attention function is computed on a set of queries that are packed into the matrix Q at the same time as with the matrices K and V. See 2.2.

$$Attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{2.2}$$

This attention function is then used in the system's multi-head attention block. The queries, keys and values are linearly projected h times with separate, learned linear projections. On each of the projections, the attention function is performed in parallel. The outputs are concatenated and projected again, which gives the final values, as one can see in 2.3 which details the function.

$$\begin{aligned} MultiHead(Q,K,V) =& concat(head_1,...,head_h)W^o \\ \text{where } head_i =& Attention(QW_i^Q, KW_i^k, VW_i^v) \end{aligned} \tag{2.3}$$

Multi-head attention is used in three ways. The standard encoder-decoder attention mechanics is seen in sequence to sequence models, self-attention layers in the encoder, and self-attention layers in the decoder.

All this amounts to that, with natural language as input data, a transformer can identify the context that confers the meaning of the word. As such, it can process a sentence out of order. The attention operation is what provides context for any position in the input sequence. Training on much larger datasets is now possible, and this led to pre-trained transformers like Generative Pre-trained Transformer 2 (GPT-2).

2.2.1 Generative Pre-trained Transformer 2

Generative Pre-trained Transformer 2 (GPT-2) is an open-source project created by OpenAI (Radford *et al.* [13]). GPT-2 can translate text, answer questions, summarize passages and generate text that is sometimes indistinguishable from human-written text. The architecture is an implementation of a deep neural network transformer model.

The project demonstrated that one no longer needed supervised learning on task-specific datasets for question answering, reading comprehension, and machine translation. GPT-2 was trained on a dataset of millions of web pages called WebText. The team behind GPT-2 wanted to show that in the future, it would be possible to achieve a state of the art performance from a generalist program. By using a large enough and diverse enough dataset GPT-2 was able to achieve a state of the art performance on 7 out of 8 tested language modelling datasets. In zero-shot settings GPT-2 performed strongly in reading comprehension and translation. It performed competitively with supervised baselines. The performance on tasks like summarizing was rudimentary. It showed a bias to recent information in the articles and would mix up information like how many cars were involved in a crash.

One of the primary uses of GPT-2 is in generating synthetic text based on an input prompt. This has been in projects such as AI Dungeon, a text-based adventure game powered by GPT-2 that will generate the storyline for the player, based on their input (Walton [14]). One reviewer described playing the game as "doing improv with a partner who is equal parts enthusiastic and drunk." (Bird [15])

2.2.2 Generative Pre-trained Transformer 3

Generative Pre-trained Transformer 3 (GPT-3) (Brown *et al.* [16]) is the newest release and improves on GPT-2 in a few significant ways. Whereas GPT-2 has 1.5 billion parameters, GPT-3 has 175 billion parameters. This was at the time of release ten times more than any competing solution and over 100 times more than GPT-2. This enabled GPT-3 to reach competitiveness on several NLP tasks with state of the art fine-tuned approaches. Brown *et al.* [16] define what they call Fine-Tuning, Few-Shot, and One-Shot. Fine-Tuning is when one updates the weights on the pre-trained model by training on a supervised dataset specific to the desired task. Few-Shot is where the model is given a few demonstrations of the desired task at inference time, but with no updates to the weights. One-Shot is the same as Few-Shot, but with only one example. On the LAMBADA dataset, which tests the modelling of long-range dependencies in text, GPT-3 improves on state of the art with Zero-Shot, One-Shot and Few-Shot.

This is not to be confused with Few-Shot or One-Shot *learning*, which refers to the problem of training a model with few or one training examples.

2.3 Convolutional Sequence to Sequence Learning

Gehring *et al.* [17] introduce convolutional sequence to sequence learning. Convolutional networks are a kind of neural network wherein the hidden layers include layers that perform convolutions. A *convolution* is defined as a mathematical operation on two functions that produces a third function. RNNs have to maintain a hidden state over the entire past of a specific sequence to deal with sequential data. This makes parallel computation impossible within a sequence. Convolutional networks do not depend on previous time steps' computations, which gives them a distinct advantage in training time. In a multi-layer convolutional neural network, a hierarchical representation is created of the input sequence. Nearby input elements interact at lower layers, and distant elements interact at higher layers. This structure gives a shorter path to capture long-range dependencies than RNNs. Gehring *et al.* [17] are able to capture relationships within a window of n words by applying $\mathcal{O}(\frac{n}{k})$ convolutional operations for kernels width k, versus the linear $\mathcal{O}(n)$ of RNNs. Because inputs to a convolutional network are fed through a constant number of kernels and non-linearities, an RNNs amount of operations scales with the length of the sentence. Therefore learning is faster on the convolutional network.

2.4 Hierarchical Neural Story Generation

Fan *et al.* [6] built a story generator that attempts to improve structure and coherence by hierarchical generation from a textual premise. The system first generates a base premise which is later used to generate a larger passage of text. The hierarchical structure ensures that the generated story has more consistency than non-hierarchical approaches, where the generator can tend to "lose the plot". Conditioning the prompt when generating the story gave more consistent results and kept the generator from drifting off-topic.

Human judges preferred the hierarchical generated stories over non-hierarchical by a factor of two to one [6]. To make sure there was a stronger connection between the generated story and its corresponding prompt, a fusion mechanism was used that trained their model on top of a pre-trained, sequence to sequence model. Language models generate on a word by word basis, so the challenge was to introduce a high-level plan as human authors typically have. They modified the cold fusion mechanism of Sriram *et al.* [18] by combining two sequence to sequence models where the hidden state of the pre-trained sequence to sequence model and training sequence to sequence model (ht) are concatenated (gt).

The stories are generated from the models using a top-k random sampling scheme. At each timestep, the probability of each word in the vocabulary being the next word is calculated. From the k = 10 most likely words, a random word is sampled. Subsequent timesteps generate based on previously selected words. This was found to be better than beam search, which often produced common phrases and repeated text from the training set.

Compared to Nearest Neighbour Search (kNN), which does a good job of generating stories that human judges can easily link to their original prompts, the hierarchical model can match kNN's performance in terms of relevance to the original prompt. Where the hierarchical model shines, it can generate an unlimited amount of stories where kNN cannot generalize from the training data.

When checking the novelty against kNN and the baseline convolutional sequence to sequence model, the hierarchical model outperforms both. The average longest common subsequence the generated story has with the training set is 8.9 for the hierarchical method, 10.2 for convolutional sequence to sequence and all 150 words for kNN.

One weakness of this system was that the generated prompts were generic compared to human prompts.

Chapter 3

Related Work

In this chapter a rundown of the history of story generators as well as a state-of-the-art of story generation will be presented. Some of the paragraphs in this section are taken from the pre-study (Staff [5]).

3.1 Story Generators

This section will give a short overview of existing story generation systems. In the past several different approaches have been taken to generate stories. Some generators have used templates, some use narrative patterns, and some develop narrative constituents into literary discourses by brute force.

3.1.1 TALE-SPIN

One of the first computational story generators was presented in Meehan [19]. It simulated rational behaviour by characters in a story world. The simulator consisted of three components. A problem solver, an assertion mechanism, and a static description of each place in the story world. The problem solver took a goal and produced subgoals and events. The assertion mechanism took an event and added it to memory. The static description described what the story looked like in a specific instant, including which relationships existed between characters. As soon as an event was asserted the consequences of the event would be computed and asserted and so on. One type of consequence was a goal. A goal was asserted by calling the problem solver, and the cycle started a new.

Meehan [19] goes on to show the challenge of making sure TALE-SPIN characters were able to understand enough to make the stories coherent. In one example, character one gives character two precisely what they want, but character two still threatens character one to give up the goods in the following sentence. This was because character two had not understood the response from character one. In order for TALE-SPIN to work, it always had to know enough to be convincing.

TALE-SPIN worked as a top-down, goal-directed problem solver. Meehan [19] defines a *problem domain* consisting of a set of representational primitives, a set of goal states and problems expressed in terms of those primitives and the procedures used to solve the problems. One of these primitives is *plans*. An example of such a plan within the system is a *delta-act* which are defined by a goal state. For instance DELTA-PROX(x,y,z) is the *delta-act* that describes when *x* wants *y* to be near *z*. As such, there is a mapping of the physical space of characters in the system. DELTA-KNOW(x,q) is the *delta-act* where *x* wants to know the answer to question *q*.

Sigma-states are top-level goals in the characters of TALE-SPIN. These are bodily needs such as hunger, thirst, sex and rest. This *sigma-states* in general are not what the stories generated are about, but they are the first goal of the characters.

Relationships between characters are characterized by competition, dominance, familiarity, affection, trust, deceit and indebtedness. Relationships are both used as preconditions to the delta-acts and have consequences for the actions. If a character is given a gift, they will feel affection towards the giver.

With all these rules and parameters in the system Meehan [19] is able to in his words create simple, reasonable stories that are not "bizzare tales".

3.1.2 AUTHOR

Where TALE-SPIN models the minds of the characters, the AUTHOR system developed by Dehn [3] models the minds of the author as a story is written. Dehn [3] thought it unlikely that a story generator where the characters and world is specified before generation begins could easily result in an interaction that would be as interesting as the introduction of Lando in Star Wars: The Empire Strikes Back. Lando is introduced halfway into the film and has a history with one of the other characters. Dehn [3] thought a character like Lando would, by a human author writing the story, be thought up when needed, and the fact that an author's goal changes as the story generation progresses was something integral to writing stories.

The four force processes driving the generation process in the AUTHOR model are:

- Author intentionality
- Conceptual reformulation
- Reminding
- The opportunity enhancement meta goal

Author intent-based story generation is described as focusing more on making a good story than letting the characters' wants in the story decide what will happen next. The end result is based on the author's need to propel the story in the directions needed. Conceptual reformulation is the process of an author coming up with an idea to integrating it into the story world. For instance, when the author thinks of a character's characterisation and then creates an episode that illustrates this characterisation. Reminding plays two critical roles in Dehn's story generation process. As a source of relevant external material such as characters and incidents from the author's life that will be used in the story and keeping track of internal material such as narrative goals. As for the meta goal, two are defined.

The goal of achieving the current narrative goal and finding better narrative goals to pursue.

3.1.3 UNIVERSE

UNIVERSE, developed by Lebowitz [20], was created to generate soap opera episode scripts. Using complex data structures representing character data, user input, and algorithms can generate several overlapping storylines. UNIVERSE split the storytelling goals it deals with into two classes. Character goals and author goals. Where Meehan [19] and Dehn [3] each had one of these, UNIVERSE attempts to combine these ideas. The expansion of author goals controls UNIVERSE's storytelling algorithm, and character goals provide constraints on what the characters can do. The author goals instantiate high-level structures for characters, for instance, causing one character to double-cross another.

UNIVERSE maintains two precedence graphs illustrating how the various upcoming author and character goals relate to each other and previous events. An author goal is selected to expand, and this process runs recursively until it hits actual events. Once it has the actual events, natural language methods are used to generate the story.

The data structures for the story world are developed before creating characters and the characters relationships. The characters are then used to create story fragments. The data for every character includes traits and interpersonal relationships. Characters possess goals and sub-goals. The system picks goals that have fulfilled pre-conditions and develops the plot towards those goals. This way, Universe can develop several parallel plots which follow the same structure as a soap opera.

3.2 MINSTREL

MINSTREL, which is detailed in Turner [21], draws on a library of stories to generate new ones. The "author"-component uses the Transform Recall Adapt Method (TRAM) to receive solutions. The TRAM system is a collection of several TRAMs that generate the fine details of a story. For example, a query is sent to the TRAM system in which a character Bob needs to do something to a dragon, but no stories are available in the library which can be used. TRAM therefore converts the dragon to a generic monster, and this new query can retrieve a fragment from the library about Bob fighting a troll with a magic sword. The result is adapted back to Bob fighting the dragon with a magic sword which satisfies all the original requirements.

The author-component plans by using Planning Advice Themes (PAT) to provide coherent solutions. Characters are put into the system with their own goals that are secondary to the PAT and the author-component's plan. MINSTREL's design is focused on order, causality and character entities. All story developments are verified in terms of character plan, story world and emotional consistency. There is also a third system called Boredom which is a table of queries and solution signatures with their corresponding boredom value. When a query is given to the TRAM system and a solution returned, the boredom value of that solution is incremented. High boredom solutions are not permitted, so in those cases, other solutions must be found. This keeps the results varied. This is a system certain artists who keep making the same thing over and over might find helpful.

3.2.1 Brutus

Brutus, another example of a grammar-based system, was a system that generated stories of betrayal as detailed in Bringsjord *et al.* [22]. It used a thorough knowledge model for representing the concept and implications of betrayal. Using a grammar-based generation model and a literary beautifier, it generated high-quality texts that compare to texts written by humans. In both Brutus and GESTER story, grammars constitute the formal representation of the structure of the kind of stories the generators rely on.

According to Bringsjord *et al.* [22] Brutus was designed to fulfil the "seven magic desiderata of storytelling". According to them, these are:

- Give proposed rigorous accounts of strong creativity a run for their money.
- Generate imagery in the reader's mind
- Situate the story in "landscape of consciousness."
- Mathematize concepts at the core of belletristic fiction.
- Generate genuinely interesting stories.
- Tap into the deep, abiding structures of stories.
- Avoid "mechanical" prose.

In Brutus, a scenario is generated based on a thematic description. A story outline is then generated by using sentence grammars from a collection. The technical architecture is split into two levels, the knowledge level and the process level. The knowledge level contains data needed to generate a story based on a given theme. The process level contains processes needed to generate the story. The knowledge level is further split into domain knowledge, linguistic knowledge, and literary knowledge. The process level is further split into thematic concept instantiation, plot generation, story structure expansion, and language generation.

3.2.2 MEXICA

MEXICA developed by Pérez y Pérez and Sharples [2], generates stories about the Mexica people. The core idea of MEXICA is to mirror an author's writing process of switching between engaged and reflective states. The generator uses a set of previous stories and a set of primitive actions (that have pre- and post-conditions) to draw from when generating the story. The engaged state builds on the current story state by picking from possible actions that work in context to create the next story beat. If it encounters an impasse or reaches the end of the tale, it enters a reflective state where it looks over the whole story, checking that pre- and post-conditions are upheld. MEXICA also encodes the characters feelings towards each other as well as the tension each story state and action will have on the reader. This is used to ensure MEXICA tries to create riveting drama.

MEXICA uses associative structures to match atoms in memory to get a set of possible subsequent actions in an engaged state. These structures are built using previous stories and the set of primitive actions fed into the system before starting the generation. The structures are dynamic because if no atom can be matched, they transform themselves to try to match another atom. Each point in the story is called a context.

The first *associative structure* is created by copying all the *emotional links* and *tensions*. MEXICA generalises the context (by removing the specific characters). It then looks in long term memory for other instances of this context. Once it has found possible next actions in this context and chosen one, it substitutes back in the specific characters and writes the next line. In one operation mode, the next action is chosen randomly. In another mode, the options that do not contribute to story development are filtered away first before selecting at random from the remaining ones.

During reflective state, MEXICA performs four primary tasks. It checks and solves preconditions for each primitive action, breaks an impasse it might have run into during the engaged state, evaluates story progress, and produces guidelines for the engaged state. The reason checking preconditions is needed is that they are not checked during the engaged state. For each story step, the reflective state verifies that all the preconditions for the action are satisfied. If they are not, an action to fulfil them is inserted, and the cycle starts again from step one. When all preconditions are satisfied, MEXICA updates the characters contexts and post-conditions, then moves on to do the same from step two. The system has two ways it evaluates the stories. Firstly how similar the story generated is to any of the previous stories. Secondly, how the tensions on the reader in the generated story compares to the tensions in the previous stories (higher tensions should mean more interesting).

3.2.3 GESTER

GESTER (GEnerating STories from Epic Rules) was an attempt at utilising grammars as representational tools to capture the structure of complex symbolic sequences as detailed in Pemberton [23]. The so-called epic rules were information about story structure in simplified narrative grammar that captured characters and events from the Old French epic sub-genre.

GESTER was written in POPLOG Prolog and used Prolog's grammar rule facility. The program accesses data on story structure which comes as a part of the narrative grammar, which, together with a database of objects and attributes, is used in generation.

A story world rules module contains actors and possible actions the GESTER system can use. During the scenario "conquest of a city", some examples of possible motivating acts are "sight of the city" and "hearing its defenders are absent". In a situation where a wife is the object, a subject may be motivated by "hearing a song about the woman".

Story world rules also contain the restrictions on possible combinations when it comes to actors and actions.

3.3 Combining Full Plots Into Multiple-plot Stories

A widespread way of creating a suspenseful story is to combine several plot lines. In Concepción *et al.* [24], they attempt to find out how it could be possible to combine several different plot lines into a single linear sequence that works as a plot. Four strategies were tried, which are detailed further down.

There is a link between a story and a plan, which has led many computational story generators to use AI planning solutions. TALE-SPIN (Meehan [19]) is such an example where the generation of the story was based around the single goal of the main character, which meant that TALE-SPIN created single-plot stories. AUTHOR (Dehn [3]) added to the characters goals by also modelling the author's goals. The author will have different goals than the character at times, such as adding conflict to the story to increase how exciting the story is. This could sometimes result in additional plot lines. Universe had several characters with different goals but no specific way to manage multiple plots.

Plots were considered input to be combined into multiple-plot stories. The inputs where created by a plot generator that used an inner template repository for managing the plot template catalogue. To construct complex stories, two or more simple plots were combined into subplots in a larger story.

The two ways of combining a plot are by either weaving scenes together where any individual scene in the combined plot was an individual scene in the original simple plots or by merging scenes together, thereby creating new scenes. Only scene weaving was used.

The four methods of combining plots in this system are

- Random
- Alternating juxtaposition
- Communicating vessels
- Chinese boxes

In *random weaving* the next plot, from which to take the next scene, is selected at random. There is no checking pre- or post-conditions for consistency.

In *alternating juxtaposed weaving* the plots to be combined are processed sequentially in a cycle. For every iteration, a scene is selected and placed next without checking pre- or post-conditions. This results in a fixed pattern. In *communication vessel weaving* the plots to be combined are again processed sequentially in a cycle. Each iteration selects a scene but only adds it if it is compatible with the story so far in terms of pre-conditions. The system tries to pick the following scene from a different plot, but it will continue with the next scene in the same plot if it can find nothing that fits it. In *Chinese boxes weaving* one plot is selected to be the main plot. The remaining plots are processed sequentially. For each additional plot, the algorithm tries to fit it between two scenes of the main plot based on if pre- and post-conditions are compatible.

The first two strategies are used as the baseline against which to measure other strategies. The result of this test was not that surprising. The weakness of random and alternating juxtaposition was that they generated instances where, due to not checking pre- or post-conditions, glaring inconsistencies were introduced, such as a character doing something after they died. Communicating vessels introduced inconsistencies by only checking pre- and post-conditions on boundary scenes. Both Chinese boxes and communicating vessels had instances of the subplot not helping the main story.

3.4 Modelling Suspense

Suspense is not well understood computationally. In Wilmot and Keller [4] they look at two ways of modelling suspense. *Surprise* which they define as looking backwards on the story to gauge how unexpected the current state is, and the other is *uncertainty reduction* which looks forward to check how unexpected the continuation is. To do this Wilmot and Keller [4] used a hierarchical language model that encodes stories and computes surprise and uncertainty reduction. This is evaluated against short stories that have been annotated by humans for their suspense.

Both surprise and uncertainty reduction can be computed directly over the story representation or indirectly over probability distributions over story representations. With a hierarchical language model based on Generative Pre-Training, they encode representations on the story level and develop an inference scheme that can use the encoded representations to compute surprise and uncertainty reduction.

Their stated goals were to test if specific psycholinguistic and economic theories could be used to correctly model suspense as a human would experience it in a story. To test their trained system, a small portion of the Writing Prompts dataset from Fan *et al.* [6] was annotated for suspense manually by humans. After experiments, it turned out to be possible to model suspense. Their model was also tested on the TRIPOD dataset from Papalampidi *et al.* [25] where it performed well in predicting turning points (which are annotated in the TRIPOD dataset).

3.5 Evolutionary Algorithms

Evolutionary computation techniques can be an excellent tool for procedural story generation as seen in Fredericks and DeVries [26].

Evolutionary computation is a set of algorithms used for global optimization inspired by biological evolution. An initial set of possible solutions is generated and then iteratively updated. Less desired solutions are removed stochastically, and as in biology, where the population gradually evolves towards fitness, this evolves towards the fitness algorithm of the function. The challenge in using this for story generation is in transforming a story in a natural language into something that evolutionary computation can solve.

Fredericks and DeVries [26] detail a conceptual procedure for genetically improving grammar-driven procedural story generation. They use a system called Tracery, which is an open-source tool that can create generative text using a grammar-based system. Tracery can define a grammar that can generate a small video game scenario that they call storylet. This is then improved with grammatical evolution. In this case, they propose to use novelty search in order to find new grammars. The goal of novelty search is to generate diverse solutions.

In Wang *et al.* [27] they present a methodology for transforming a written story in event-level and hierarchical-level grammar using a network representation. The result of this methodology is a set of formal entities consisting of a hierarchy of interdependent events with characters, time, space, a description, a topic, and an object that can be transformed into a chromosome and vice versa. This can be used for evolutionary computation techniques that develops a story using subjective metrics such as coherence, novelty, "interestingness", and quality—and also using objective metrics such as the logical structure of events and participant arrangement to control the result.

The TimeML ISO standard in Pustejovsky *et al.* [28] for annotating events from text defines an event, a cover term for situations that happen, occur, hold, or take place which can be punctual or last for some time and also includes those predicates describing states or circumstances in which something obtains or holds true.

To make it easier to recognize an event Wang *et al.* [27] relaxed the constraints of the TimeML definition. An event is a predicate that denotes an action, state or occurrence in a story. It has a position in time and space in the story world and has participants as parameters. Events are then sorted into a dependence network where each node is an event in the story, and directed lines denote a dependent relation from an enabling condition.

A Prolog program is implemented to automatically group events. A chain relation data structure is created to represent the story dependence network. This is important to generate both forward narration and flashbacks. A story narration is then represented into an evolutionary computation genotype and phenotype with a flashback as a permutation. To find a permutation of the chains and participants in the dependence network is a travelling salesman problem. The genotype to phenotype mapping transforms the permutation genome into a valid story narration. The steps to get a text from a genotype-phenotype mapping procedure are first to scan the genome to get a list of chain genes in order left to right. Then extract the participant arrangement from the participant list before enumerating the text representation of each of the chains one after another in the order they had in the chain.

Chapter 4

Data

Generating text is a Natural Language Processing (NLP) problem. For a neural network-based story generator, a training dataset is required. This should preferably be large. This project aims to generate a twist, which means training on the dataset has to result in a system that will generate a twist. In this chapter, an overview of possible sources of data to build datasets will be provided as well as an analysis of their suitability for this project. Furthermore, existing datasets for different story related NLP training are discussed. Finally, the plans for creating a dataset for this project are detailed.

4.1 Knowledge Sources

In this section, an overview of good sources of story data is given. These sources are helpful for different tasks as they provide slightly different benefits. The information on the sites is publicly available, and in terms of being scraped, the sites are either worried about the impact on their servers, or about people trying to make clone sites. They are not worried about training on the data for research.

4.1.1 Reddit - Writing Prompts

Reddit[1] is a social news aggregation, web content rating, and discussion website. Users can submit links, text posts, images and videos. Each post has a comment section, and comments can, in turn, be upvoted and downvoted, so the most interesting content is always sorted to the top, both in the top-level forums and in the comment section of each post. Users can create their own Reddit forums dedicated to whatever they please, which are called subforums. One such subforum is called *Writing Prompts*. In the Writing Prompts Reddit subforum, the idea is that a poster makes up an interesting writing prompt such as "[WP] When you die, you appear in a cinema with a number of other people who look like you. You find out that they are your previous reincarnations, and soon you all begin watching your next

[1]`https://www.reddit.com/`

life on the big screen.". The top-level comments on this post will be short stories based on this prompt written by the users. The comments on the stories will be the user's reaction to these stories. As with everything in Reddit, all of this is ranked by the users upvoting and downvoting. In theory, the best story will be the first one at the top beneath the post. Because the Writing Prompts subforum has been going strong for years with at the time of writing over 15 million users, it has amassed many prompts and stories, all sorted by the users from best to worst. The prompts here have a large diversity of topics. To use this as a knowledge source for this project, there would have to be enough stories featuring twists and a way to sort these from the rest. There have been a few threads where the prompt given is to generate a story featuring one or more twists, which could be used to supplement other sources.

4.1.2 FanFiction.net

FanFiction.net[2] is a site where people can post fan-created stories. This means stories set in existing popular universes often using famous characters. This turned out to be something very popular, and the site now has over 12 million registered users and over 800 000 Harry Potter fan-written stories alone, Harry Potter being the most popular universe on the site. The site also hosts one of the longest works of fiction ever written, clocking in at over 4 000 000 words and counting at the time of writing. As a repository of human-written story text, FanFiction.net is huge in terms of the number of words and stories. However, as mentioned, a dataset created from Fanfiction will skew towards Harry Potter and likely the romance genre as this is very popular in Fanfiction. Unfortunately, stories are not tagged in any detailed way relating to content, which makes it hard to use for this project.

Archive of Our Own

Archive of Our Own[3] is a non-profit, open-source repository of fan fiction. It now hosts over 7 million works in over 40 000 fandoms. In addition to being similar to FanFiction.net Archive of Our Own has a tag system where writers add tags to their stories similar to hashtags on twitter. This could be very useful when trying to build certain datasets for training as you can filter the stories on almost anything and find collections of stories featuring quite specific elements. Unfortunately, not on plot twists.

Wattpad

Wattpad[4] is a website and app where user can publish their original stories. The site consists of user-generated stories similar to FanFiction.net and Archive of Our Own, but without the copyrighted elements of existing characters. Wattpad is

[2]`https://www.fanfiction.net/`
[3]`https://archiveofourown.org/`
[4]`https://www.wattpad.com/`

more serious about allowing its users to get their work noticed and published or bought and adapted in some cases. In terms of building a dataset, it would be more ideal than FanFiction.net, which has too much Harry Potter content for the purposes of this project. Wattpad also has tags, but not for plot twists.

4.1.3 TV Tropes

TV Tropes[5] is a wiki filled with articles on different "tropes" which is defined in this context as "a common or overused theme or device: CLICHÉ" (Merriam-Webster [29]). As such, it is possible to pick out articles with examples of plot twists and scrape these. An example would be the article called "The Reveal", defined as "the audience is given new information which had been withheld to create suspense"[6]. This article contains examples from works of film, animation, literature of The Reveal, which can be trained on in the same way as prompts from Writing Prompts were trained on in Fan *et al.* [6]. This was the first promising compilation of twists this project was able to uncover. The problem is that even with a huge list of twists, many of them are not explained very well in the articles. TV Tropes has a very relaxed view on the formatting of entries, and as such, some entries are full paragraphs explaining the set-up and twist, which is helpful to the project. Others are one sentence such as: "Emily is the killer". In that case, a neural network will not learn much about what makes a twist. The set up which made it seem like Emily was not the killer, is missing.

4.1.4 Wikipedia

Wikipedia[7] has a detailed plot synopsis of most known books and movies. Most story generators are designed only to build a coherent plot over a few paragraphs, and this is precisely the format of these plot synopses. The plot synopses at Wikipedia are an excellent resource to train a neural network on real stories that are not too many words. This is especially interesting if it is possible to filter the stories somehow for stories involving twists. Thanks to IMDb, for movies, it is possible to do this.

4.1.5 IMDb

IMDb - Internet Movie Database[8] is a database of information about films, television programs, and more. It is mostly used to check cast, production crew, user ratings, fan and critical reviews. However, it also features short plot summaries as well as tags for plot elements and genres. The tags are added by users and rated for how relevant they are by users. The movie Blade Runner 2049 (2017)

[5]https://tvtropes.org/
[6]https://tvtropes.org/pmwiki/pmwiki.php/Main/TheReveal
[7]https://www.wikipedia.org/
[8]https://www.imdb.com/

	Source (prompts)	Target (stories)
Lines	272 600	272 600
Words	6 440 010	139 699 307

Table 4.1: Size of Writing prompts dataset, the lines correspond exactly

has 563 tags[9] including "sequel", "artificial intelligence", "cyberpunk" and "memory implant". On the "artificial intelligence" tag, there are 803 titles. Examples include Westworld (2016-), Avengers: Endgame (2019), Interstellar (2014) and The Matrix (1999). If a project wanted to compile a movie script featuring AI, IMDb would be a good place to find a list of films featuring this trope. This also extends to plot twists, which is of benefit to this project. Almost 1000 films with a twist in the story are listed under the first tag. Unfortunately, many of these are very obscure, and the twist is also not a very big part of the story necessarily. To be of any use for supervised training, a Wikipedia counterpart must exist. The program written to compile these datasets would have to take this into account. More on that in chapter 5.

4.2 Existing Datasets

Earlier neural network-based story generation projects have used several datasets. This section will explain what has been made in terms of these datasets and how they have been used.

4.2.1 Writing Prompts

Fan *et al.* [6] developed a dataset based on Reddit's Writing Prompts forum. They scraped three years worth of prompts using the official Reddit API. The dataset consists of 272 600 prompts in one file as the source and one file consisting of a story from the comments of each prompt to be used as the target. The source and target file have the same amount of lines, and the prompt in the source corresponds to the story on the same line in the target. See table 4.1. As such, one has an excellent dataset for supervised learning, and Fan *et al.* [6] can teach the system to map a prompt to a story.

The dataset also features test stories and validation stories, accounting for 5% of the scraped stories (approx 15 000 stories each for test and validation). They made sure the stories were at least 30 words, which is of note since it indicates a reasonable lower limit for the target dataset for this project. In terms of usability for this project, this dataset, when used with Fan *et al.* [6]'s system, gives a good baseline of high performance to compare to the results for this project.

[9] `https://www.imdb.com/title/tt1856101/keywords?ref_=tt_stry_kw`

Modeling Suspense

Wilmot and Keller [4] wanted to model suspense in a story. To create a dataset for training, they decided to annotate a portion of the Writing Prompts dataset from Fan *et al.* [6]. Five people annotated 100 stories each. Every sentence was annotated for either increasing, decreasing or not changing the level of suspense. Suspense can be closely tied to a twist as the reveal often functions as the peak of suspense in a twist story. However, a dataset annotated to model the suspense of stories is not helpful to teach a system to generate twists. 500 entries are also much fewer than needed.

4.2.2 TRIPOD

Papalampidi *et al.* [25] wanted to analyse movie plots by identifying turning points. Turning points can, in some cases, be the reveal of the twist, but as turning points are defined by Papalampidi *et al.* [25] twists would be a minority of them. They modelled five kinds of turning points.

- *Opportunity*, an introductory event.
- *Change of plans*, an event where the main goal of the story is defined.
- *Point of no return*, an event where the main character is forced to commit to their goal.
- *Major setback*, an event where everything falls apart.
- *Climax*, the final big event of the story.

To do this, they created the TRIPOD dataset, which contains 99 screenplays over a variety of different movie genres, narrative types and that closely resemble the finished movies. These were annotated for all five defined kinds of turning points at the synopsis level. Because twists are often a specific type of turning point, this was interesting, but the problem again was that the TRIPOD data was both too short and not specific enough to train a system to generate a twist.

4.3 Compiling Datasets for This Project

Since no suitable dataset could be found, several new datasets for this project had to be created. To fully explore the possibilities of generating a twist, many approaches had to be tested—supervised learning with different datasets and fine-tuning a pre-trained model. To find a source of twists large enough was an enormous hurdle that was never fully solved to this project's satisfaction—more on that in chapter 7. The two primary datasets used were two original datasets created for this project which will be called the *TV Tropes dataset* and the *The Twist dataset*. The TV Tropes dataset consists of one file of prompts scraped from TV Tropes and one file of stories generated by GPT-2. These were used together for supervised learning with Fairseq. The TV Tropes prompts were also used alone to fine-tune an implementation of GPT-2. Because the TV Tropes dataset was shorter than needed and partially generated, an IMDb-Wikipedia dataset was also compiled and appended onto the TV Tropes dataset. This is called The Twist Dataset and is of higher quality than the base TV Tropes dataset. The Twist dataset was also tested with Fairseq and with GPT-2 Finding all this information, scraping it and formatting it perfectly for training was an enormous undertaking which is detailed in chapter 5.

Dataset	Lines	Words	Human Written	Summary
Writing Prompts Dataset	**Source:** 272 601 **Target:** 272 601	**Source:** 6 440 010 **Target:** 139 699 307	✓	Created by Fan *et al.* [6] and features a very large dataset of prompts and corresponding human written stories for supervised learning.
TV Tropes Dataset	**Source:** 1 267 **Target:** 1 267	**Source:** 62 066 **Target:** 248 774	**Source: ✓** **Target: ✗**	A dataset of twist-like prompts created for this project by scraping certain TV Tropes articles. The corresponding stories for supervised learning were generated using GPT-2
The IMDb Dataset	**Source:** 240 **Target:** 240	**Source:** 6 572 **Target:** 139 960	✓	A dataset containing prompts scraped from IMDb of movies featuring the tag "plot twist". The corresponding target stories are scraped from Wikipedia which features longer plot synopses.

The Twist Dataset	**Source:** 1 507 **Target:** 1 507	**Source:** 68 638 **Target:** 388 734	**Source:** ✓ **Target:** Partly	A concatination of the TV Tropes dataset and the IMDb dataset. The IMDb dataset while very nice, was far too short.

Table 4.2: Datasets created for and used in this project. The Writing Prompts dataset is there as reference

Chapter 5

Architecture

Out of all the projects detailed in chapter 3 the system with the most promise at being able to generate a twist is the Fairseq system used in Fan *et al.* [6]. This is because it uses a model that takes great lengths to improve the relevance of the output story to the input prompt by using a hierarchical model and model fusion. Recurrent and convolutional networks have no problems modelling sentences. However, their performance is not as strong when modelling several paragraphs, which keep their relevance to the prompt all the way through. A twist is based on the stories high-level structure. The audience is fed a premise along with the possibility of several red herrings. Then this is subverted. Fan *et al.* [6] includes the ability to plan by splitting the generation into two levels. Therefore, the project can be used to generate a twist. This chapter gives a detailed view of the tactics and architecture used in this project. The first section will go in-depth about building the dataset. The second will explain the details surrounding the fairseq system. The third will explain how GPT-2 was implemented.

5.1 Preparing TV Tropes Dataset

To build the TV Tropes dataset, several TV Tropes articles had to be scraped for their entries and cleaned up carefully. A scraper written in Python was built. A custom scraper was written to scrape the data, which first picked out all text on the page. The data was then pruned using regex to delete bad lines.

Code listing 5.1: The script used to scrape data from TV Tropes.

```
for site in sites:
    page = requests.get(site)
    soup = BeautifulSoup(page.content, 'html.parser')
    results = soup.get_text()
    all_sites.append(results)

with open("scrapeddata.txt", "w", encoding='utf-8') as f:
    for line in all_sites:
        if line.isalnum():
            continue
        f.write(line + "\n")
```

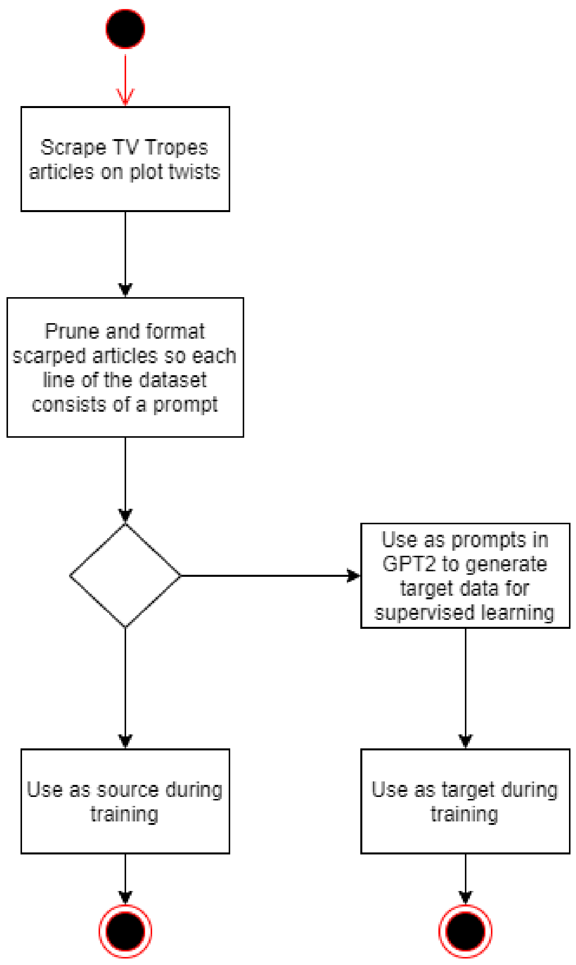

Scrape TV Tropes articles on plot twists
Prune and format scarped articles so each line of the dataset consists of a prompt
Use as prompts in GPT2 to generate target data for supervised learning
Use as source during training
Use as target during training

Figure 5.1: Control flow for creating the TV Tropes dataset

The TV Tropes dataset consists of the entries of several articles on "twist like" tropes. These articles contain many examples of twists that function as prompts in the architecture. To clean them up further, short entries had to be removed as they were useless for being used as a prompt.

Articles from TV Tropes could be added to the list of sites, and a file with the text on the page would be created. To work well with Fairseq this data had to be in the form of one "prompt" per line. As the scraper took down all text, the file had to be edited to become a dataset. For a comparison between the Writing Prompts dataset and the TV Tropes dataset see table 5.1.

Ideally, there would be far more twist examples to train the deep neural network, but this is the most this project could find. Some comparisons to results with the Writing Prompts dataset will be made to show the weaknesses.

In the Writing Prompts dataset, the target stories used for supervised learning

	WritingPrompts	TV Tropes
Source	6 440 010	62 066
Target	139 699 307	248 774

Table 5.1: Size comparison between TV Tropes dataset and Writingprompts dataset counted in number of words

were all written by humans, but target stories did not exist in the case of TV Tropes. To create target stories, GPT-2 was used on the prompts to generate text which could serve as the target (see listing 5.2).

Code listing 5.2: Python script for GPT-2 generation.

```
# module load Python/3.8.6-GCCcore-10.2.0
# source sourcegen/bin/activate

from transformers import pipeline
story_generator = pipeline("text-generation",
                           "pranavpsv/gpt2-genre-story-generator")

# Input format: <BOS> <genre> Optional prompt...
# Supported genres: superhero, sci_fi, horror, action, thriller, drama

with open("targetdata.txt", "w", encoding='utf-8') as g, open("scrapeddata.txt", "r
    ", encoding='utf-8') as f:
        for line in f:
            story = story_generator("<BOS>␣<sci_fi>␣" + line, max_length= 500)
            print(story)
            g.write(story[0]["generated_text"] + "\n")
```

The improvements were made to the TV Tropes dataset to help the deep neural network included removing short entries. This is because TV Tropes is a wiki, and there is little standardization in formatting and entry. Some entries are helpful to this project with a lengthy explanation of the setup and payoff of a particular twist. In contrast, others are a short sentence akin to "Mary's father was actually alive".

5.2 Preparing the IMDb Dataset

The IMDb dataset is based on entries in the database with the tag "plot twist". IMDb has a concise plot synopsis that often only includes the set-up or premise of the story. This is akin to a writing prompt. As such it can be used as a source dataset during Fairseq training. On the other hand, Wikipedia often has the entire plot summarized for movies that have an entry. This is ideal for a target dataset during training. The only problem was that IMDb had a lot of obscure projects with the plot twist tag. Fortunately, it is possible to sort them on popularity. This made it easy only to include entries popular enough to have a corresponding Wikipedia article. The IMDb synopsis for the movie The Shawshank Redemption looks like this:

> Two imprisoned men bond over a number of years, finding solace and eventual redemption through acts of common decency.[1]

Moreover, the Wikipedia plot synopsis starts like this:

> In 1947 Portland, Maine, banker Andy Dufresne is convicted of murdering his wife and her lover and is sentenced to two consecutive life sentences at the Shawshank State Prison. He is befriended by Ellis "Red" Redding, an inmate and prison contraband smuggler serving a life sentence. Red procures a rock hammer and a large poster of Rita Hayworth for Andy[...][2]

The structure with a short prompt and a related story is the same as with the Writing Prompts dataset. This can therefore be used by the same system. The page with the list of projects with the "plot twist" tag had to be scraped for the project synopses to compile the dataset. These were then used in a fuzzy search through Wikipedia to get the corresponding longer Wikipedia plot synopsis. This would enable experiments with a human-written target that is still relevant to plot twists. The control flow of the program written to do this is shown in figure 5.2.

5.3 Fairseq Controller

Hierarchical Neural Story Generation is a state of the art technique for improving high-level structure to a generated story.

A pre-trained convolutional sequence to sequence model was trained on a dataset of writing prompts to train the fusion model. The pre-trained model is provided to the second convolutional sequence to sequence model and is integrated using the fusion mechanism. The models are implemented using the Fairseq-py library in PyTorch.

To make experiments easier to set up and repeat with different settings and datasets, a controller had to be designed and written.

In figure 5.3 the control flow of the Fairseq controller is explained. It was written to make it easier for this project to run experiments using Fairseq with different settings and datasets. The datasets could be easily chosen by changing only one variable in the program, which would also determine where the output of training and generation was saved. It was also easy to determine which of the prepossessing, training, fusion or generation the program would run. This would drastically cut down on time it took to set up experiments and keep track of results.

5.3.1 Training

During the training, the hyperparameters of the Fairseq system had to be carefully chosen to get the best result. In this subsection, the settings used are listed, and

[1] `https://www.imdb.com/title/tt0111161/`

[2] `https://en.wikipedia.org/wiki/The_Shawshank_Redemption`

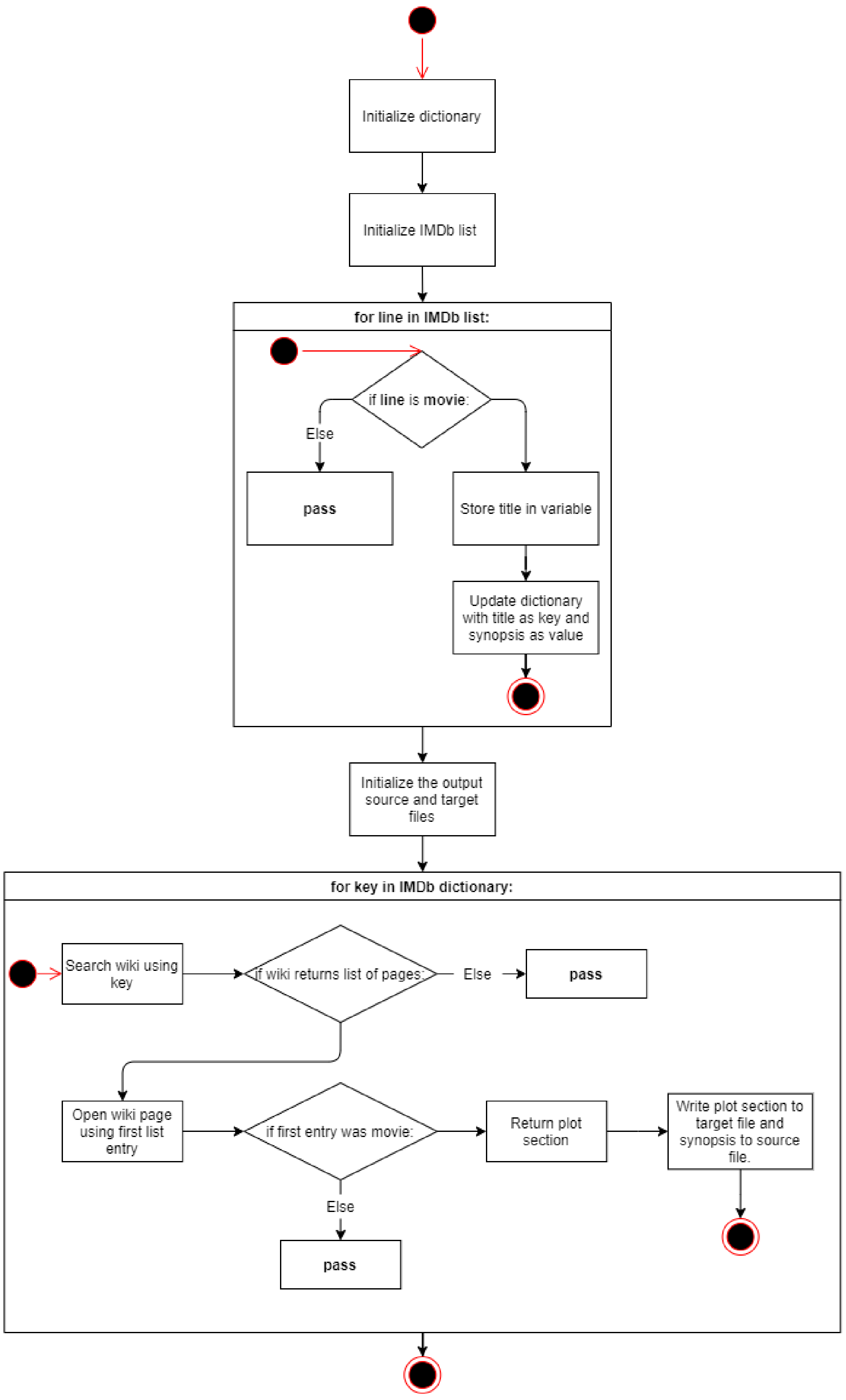

Initialize dictionary
Initialize IMDb list
for line in IMDb list:
if line is movie:
Else
pass
Store title in variable
Update dictionary with title as key and synopsis as value
Initialize the output source and target files
for key in IMDb dictionary:
Search wiki using key
if wiki returns list of pages:
Else
pass
Open wiki page using first list entry
if first entry was movie:
Return plot section
Write plot section to target file and synopsis to source file.
Else
pass

Figure 5.2: Control flow for creating the IMDb-Wikipedia dataset

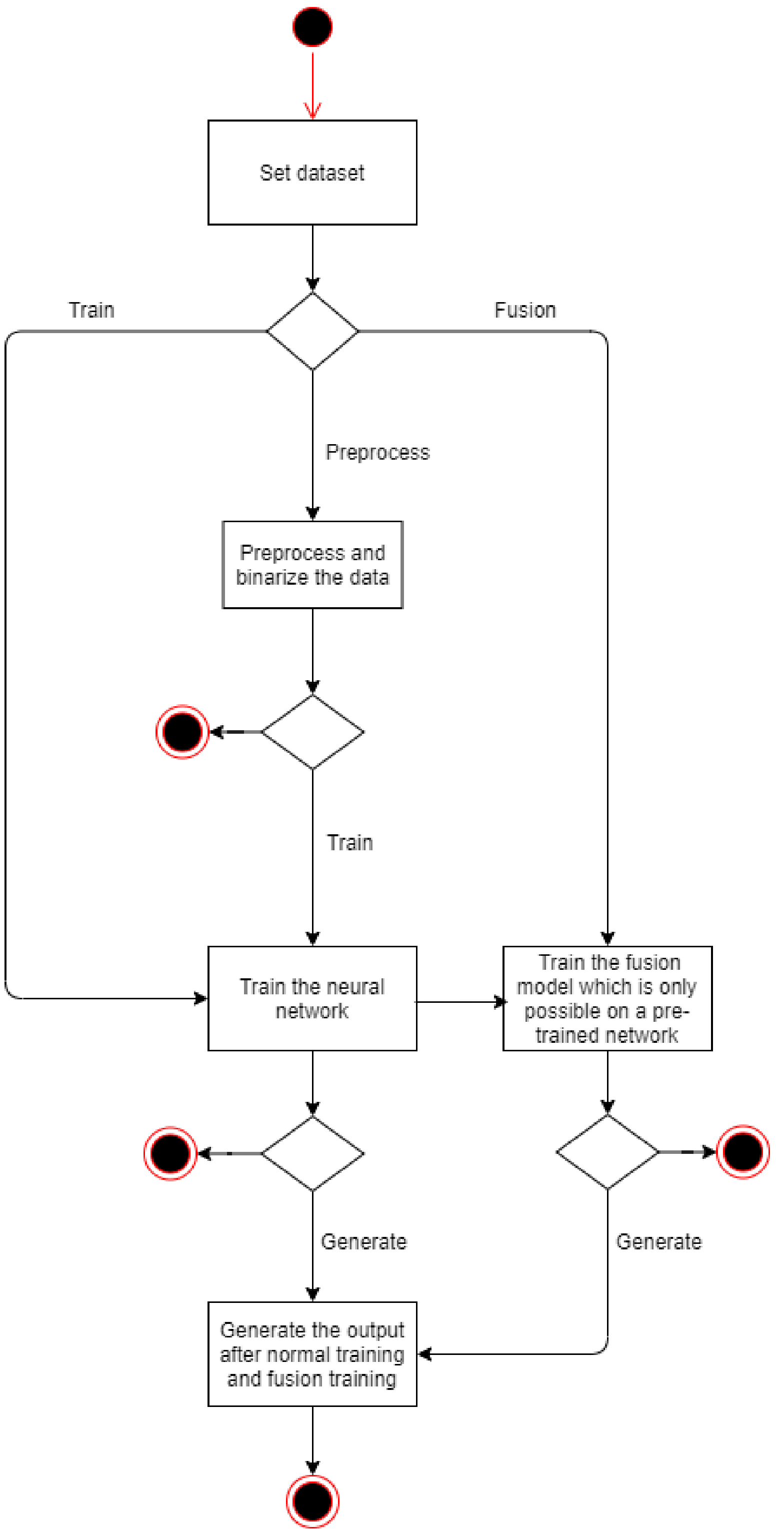

Set dataset
Train
Fusion
Preprocess
Preprocess and binarize the data
Train
Train the neural network
Train the fusion model which is only possible on a pre-trained network
Generate
Generate
Generate the output after normal training and fusion training

Figure 5.3: Control flow for the Fairseq controller

the variations are discussed.

Code listing 5.3: The hyperparameters to binarize the dataset with the settings used for the writingpropmts dataset.

```
--padding-factor 1
--thresholdtgt 10
--thresholdsrc 10
```

When binarizing the dataset in Fan *et al.* [6] padding-factor, –thresholdtgt, and –thresholds have to be chosen. This turned out to be a significant contributor to a lot of <unk> in the output, that is, unknown words. –thresholdtgt maps words appearing less than threshold times to unknown. The default argument is 0. –thresholdsrc is the same. –thresholdtgt controls the target, and one is for source. The setting of 10 is acceptable for the Writing Prompts dataset, which is large, but for the TV Tropes dataset and IMDb dataset, this ended up making far too much <unk>, and as such, for those runs, no argument was given. –padding-factor pads dictionary size to be multiple of N. The default value is 8. In this case, it is set to 1, which means no padding is done.

Code listing 5.4: The hyperparameters to train with the settings used for the Writing Propmts dataset.

```
fairseq-train data-bin/writingPrompts
-a fconv_self_att_wp
--lr 0.25
--optimizer nag
--clip-norm 0.1
--max-tokens 1500
--lr-scheduler reduce_lr_on_plateau
--decoder-attention True
--encoder-attention False
--criterion label_smoothed_cross_entropy
--weight-decay .0000001
--label-smoothing 0
--source-lang wp_source
--target-lang wp_target
--gated-attention True
--self-attention True
--project-input True
--pretrained False
```

In listing 5.4 the hyper-parameters used in Fan *et al.* [6] are listed. Some important ones to note are that –pretrained must be changed to True when training the fusion model.

5.3.2 Generation

Code listing 5.5: The hyperparameters used to generate with the settings used for the writingpropmts dataset

```
    fairseq-generate data-bin/writingPrompts
    --path /path/to/trained/model/checkpoint_best.pt
    --batch-size 32
    --beam 1
    --sampling
    --sampling-topk 10
    --temperature 0.8
    --nbest 1
    --model-overrides
    "{'pretrained_checkpoint':'/path/to/pretrained/model/checkpoint'}"
```

In listing 5.5 the standard settings for generation from Fan *et al.* [6] are listed. –model-overrides is only used for fusion model training.

5.4 Architecture of GPT-2 Target Generation

The next challenge was that they scraped Reddit's writing prompts subreddit in the Hierarchical Neural Story Generation project. This was a brilliant move because the subreddit works so that a thread is started by someone making a prompt, and then the comments in the thread will be short stories based on this prompt. This gave them a way to make a dataset consisting of prompts and their related short stories to train the system. While TVTropes made it possible to more or less build a dataset of prompts, it still lacked the corresponding stories. So in keeping with the idea of story generation, the idea became to generate these related stories using a GPT-2 based system. Going through the dataset and generating corresponding stories would make it possible to train the Hierarchical Neural Story Generation system using the new dataset.

In figure 5.4 the target generation program is explained. The resulting output from this was a file with GPT-2 generated target stories that were used in experiments with the TV Tropes dataset.

5.5 Architecture of GPT-2 Source Generation

Because generating a source twist prompt using Fairseq was less feasible than it first seemed GPT-2 was also tested. In this case, because GPT-2 should be able to give a better performance with fewer training examples, the TV Tropes and IMDb dataset might yield some interesting results. To accomplish this, a GPT-2 implementation that would take the datasets and fine-tune on them had to be built. GPT-2 needs fewer examples in the dataset during fine-tuning as it is already pre-trained. Based on Richard Bownes' Medium article *Fine Tuning GPT-2 for Magic the Gathering Flavour Text Generation*.[3] An implementation built for IDUN was created. However, this fine-tuning did not take very long, so using IDUN might have been overkill. Setting up the experiment entailed setting up an environment on IDUN by pip installing *pandas*[4], *transformers*[5] and *torch*[6] (always good to remember that PyTorch is *not* "pip install PyTorch", but rather "pip install torch").

[3]Medium article
[4]`https://pypi.org/project/pandas/`
[5]`https://pypi.org/project/transformers/`
[6]`https://pypi.org/project/torch/`

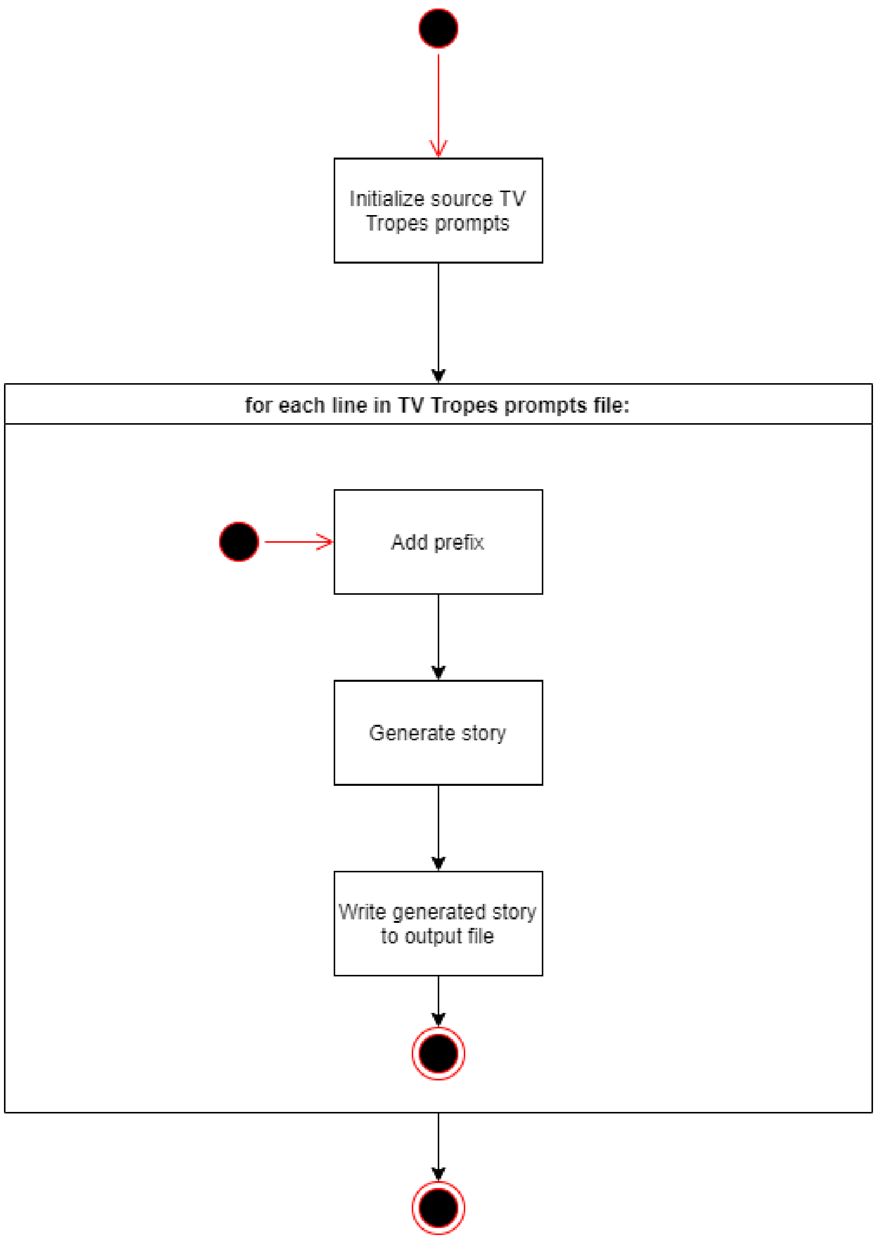

Initialize source TV Tropes prompts
for each line in TV Tropes prompts file:
Add prefix
Generate story
Write generated story to output file

Figure 5.4: Control flow for the GPT-2 target generation program

After setting this up, GPT-2 was fine-tuned by training for four epochs. Each of these took about 2 minutes.

5.6 Computation

This project performed computations related to training and generation on the NTNU IDUN computing cluster. [1] The cluster has more than 70 nodes and 90 GPGPUs. Each node contains two Intel Xeon cores, at least 128 GB of main memory, and is connected to an Infiniband network. Half of the nodes are equipped with two or more Nvidia Tesla P100 or V100 GPGPUs. Storage is provided by two storage arrays and a Lustre parallel distributed file system.

Using IDUN was a massive benefit as the computing power available to this project without IDUN would not have been enough to complete all experiments. IDUN was easy to use and very stable, and for the duration of this project, there was never an issue relating to IDUN usage.

Chapter 6

Experiments and Results

To answer the research questions, experiments were done, which will be presented in this chapter. The experiments were done on two created datasets and with a few different technologies. The first section details the plan for the experiments. The second section details how the experiments were set up. Finally, the results are shown. For Research Question 1, no experiments were carried out, as the nature of the question does not lend itself to experiments. The section detailing the findings for RQ1 will be more literature-based.

6.1 Experimental Plan

Because of the problematic nature of the research questions, in which a good result from the story generation is needed to test it on human judges, it was essential to have a plan for developing a baseline to which the experiments with the TV Tropes dataset can be compared. The Writingprompts dataset will be used to generate a baseline to compare the performance of the TV Tropes dataset and Twist dataset. If possible, the results from using the Twist dataset will also be used in a blind test with human judges. An overview of all experiments can be seen in table 6.1.

Dataset	Lines	Words	Human Written	Experiment Description
Writing Prompts Dataset Fairseq ex-periement	**Source:** 272 601 **Target:** 272 601	**Source:** 6 440 010 **Target:** 139 699 307	✓	The system was using the data and settings from Fan *et al.* [6] to test that everything worked as they claimed before using it for this project's purpose.
TV Tropes Dataset Fairseq ex-periment	**Source:** 1 267 **Target:** 1 267	**Source:** 62 066 **Target:** 248 774	**Source:** ✓ **Target:** ✗	Once the prompts had been gathered from TV Tropes, and the target stories had been generated from GPT-2 and experiment with specific hyperparameters for this project was run
The Twist Dataset Fairseq ex-periment	**Source:** 1 507 **Target:** 1 507	**Source:** 68 638 **Target:** 388 734	**Source:** ✓ **Target:** Partly	The IMDb dataset is so small that owing to the amount of <unk> in the results from training fairseq with TV Tropes dataset it will only be used together with the TV Tropes dataset to try to mitigate this.
The TV Tropes Dataset GPT-2 ex-periment	**Source:** 1 267 **Target:** 1 267	**Source:** 62 066 **Target:** 248 774	**Source:** ✓	For this experiment the source part alone is used to fine tune GPT-2 to generate twist prompts. GPT-2 needs less examples in the training data to produce English as it is pre-trained.

The Twist Dataset GPT-2 experiment	**Source:** 1 507 **Target:** 1 507	**Source:** 68 638 **Target:** 388 734	**Source:** ✓	For this experiment the source part alone is used to fine tune GPT-2 to generate twist prompts. GPT-2 needs less examples in the training data to produce English as it is pre-trained.

Table 6.1: Experiments that will be carried out for this project

The analysis will be done first by combing the output for interesting entries manually. This will indicate the quality of the TV Tropes and Twist datasets and the level of success in terms of generating a quality output. The Writing Prompts dataset is already known to produce quality output. This should be enough to give a satisfactory answer to the research questions.

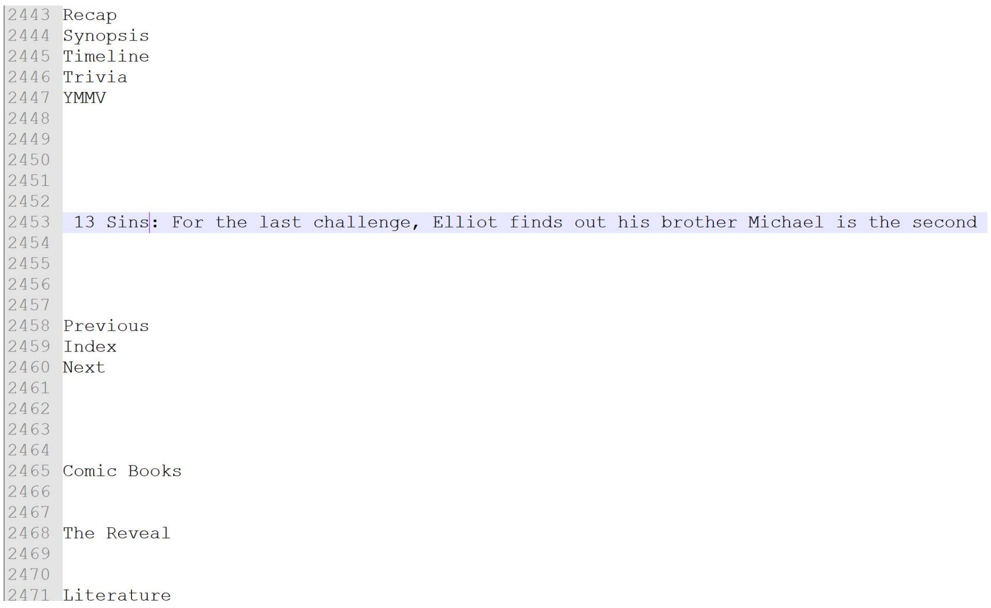
6.2 RQ1 Are There Suitable Datasets for Generating Twists?

In looking for a dataset, several approaches were taken. Looking through relevant literature and using search engines were the main ones. Datasets for particular neural network usages are not always easy to find. They exist if a more extensive project has created them for their one specific use. The dataset from Fan *et al.* [6] is one of the best in terms of training neural networks for generating stories and has been used by others. This dataset is brilliant because it includes nearly three hundred thousand prompts and human-written short stories that go with the prompts. This way, they have an abundance of prompts and are simultaneously able to create the target dataset for training. Reddit users provided the source for this dataset by submitting stories to the Writing Prompts subforum. To find something like that for twists may have solved both research questions right away. However, it turned out not to be so easy.

6.2.1 TV Tropes

The only list of twists anywhere large enough to make a dataset was several articles on TVTropes.org. As one can see in figure Figure 6.1 sometimes the very data needed was all on one line (due to shoddy HTML), and the scraped text file mainly consisted of other random text on the page.

Several regular expressions were used to edit the file in Visual Studio Code, and piece by piece, it took form. Every line had to be its own prompt, as one can



```
Recap
Synopsis
Timeline
Trivia
YMMV




 13 Sins: For the last challenge, Elliot finds out his brother Michael is the second




Previous
Index
Next




Comic Books


The Reveal


Literature
```

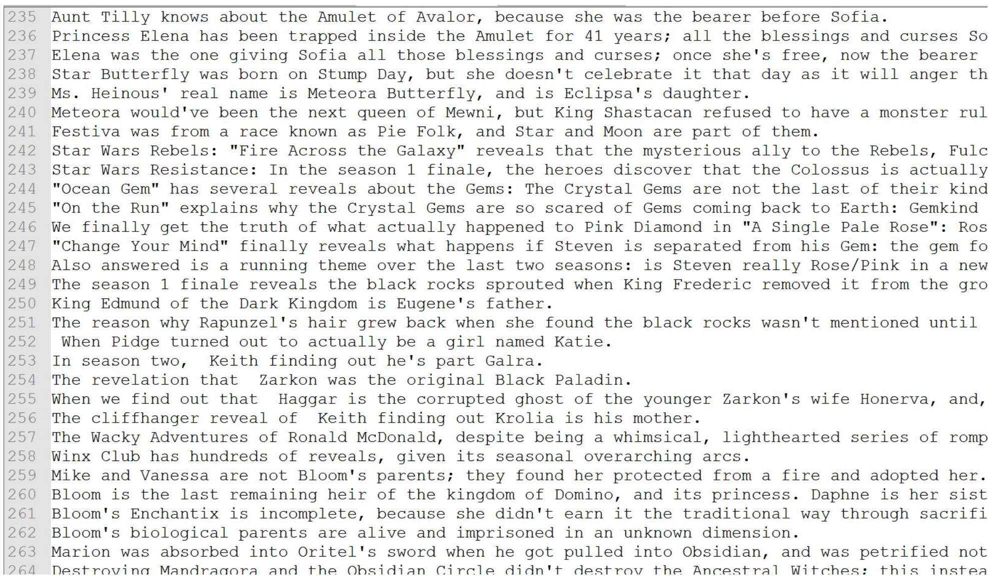
Figure 6.1: Raw data from the TV Tropes scraper. The highlighted line is the only interesting data in this. Everything else must be pruned.

see in figure Figure 6.2. To reach this level, unwanted text was pruned first. Then newlines were restored between entries where needed. The blank lines were then removed. Finally, entries under a certain length were removed. This final was done because many of the entries on TV Tropes are of low quality for this project.

To run the Hierarchical neural system properly, simply having prompts is not enough. Target data is also needed. In the case of Fan *et al.* [6] as previously mentioned, they found a straightforward and elegant solution. However, the only way to have target data for the TV Tropes dataset would be to either use the original Writing Prompts target data or generate target data based on these prompts. Both approaches were tested.

6.2.2 GPT-2

GPT-2 was used to generate a target dataset for training with the TV Tropes dataset as source. The target dataset had to consist of corresponding short stories to the prompts, or in the case of the TV Tropes dataset, corresponding twists in the source dataset. Because GPT-2 is able to take in a prompt and extend it, it was the best option for this purpose, barring having humans make it manually. GPT-2 was able to create short stories based on every line in the TV Tropes dataset. Figure 6.3 show a sample of the generated target data. As one can see on some lines GPT-2 struggled to extend it far. There was also the problem that used this way GPT-2 was unlikely to include any twist in the output. When used to extend a prompt GPT-2 keeps predicting which word to say next makes sense without any thought to story structure or getting to a point.



```
Aunt Tilly knows about the Amulet of Avalor, because she was the bearer before Sofia.
Princess Elena has been trapped inside the Amulet for 41 years; all the blessings and curses So
Elena was the one giving Sofia all those blessings and curses; once she's free, now the bearer
Star Butterfly was born on Stump Day, but she doesn't celebrate it that day as it will anger th
Ms. Heinous' real name is Meteora Butterfly, and is Eclipsa's daughter.
Meteora would've been the next queen of Mewni, but King Shastacan refused to have a monster rul
Festiva was from a race known as Pie Folk, and Star and Moon are part of them.
Star Wars Rebels: "Fire Across the Galaxy" reveals that the mysterious ally to the Rebels, Fulc
Star Wars Resistance: In the season 1 finale, the heroes discover that the Colossus is actually
"Ocean Gem" has several reveals about the Gems: The Crystal Gems are not the last of their kind
"On the Run" explains why the Crystal Gems are so scared of Gems coming back to Earth: Gemkind
We finally get the truth of what actually happened to Pink Diamond in "A Single Pale Rose": Ros
"Change Your Mind" finally reveals what happens if Steven is separated from his Gem: the gem fo
Also answered is a running theme over the last two seasons: is Steven really Rose/Pink in a new
The season 1 finale reveals the black rocks sprouted when King Frederic removed it from the gro
King Edmund of the Dark Kingdom is Eugene's father.
The reason why Rapunzel's hair grew back when she found the black rocks wasn't mentioned until
 When Pidge turned out to actually be a girl named Katie.
In season two,  Keith finding out he's part Galra.
The revelation that  Zarkon was the original Black Paladin.
When we find out that  Haggar is the corrupted ghost of the younger Zarkon's wife Honerva, and,
The cliffhanger reveal of  Keith finding out Krolia is his mother.
The Wacky Adventures of Ronald McDonald, despite being a whimsical, lighthearted series of romp
Winx Club has hundreds of reveals, given its seasonal overarching arcs.
Mike and Vanessa are not Bloom's parents; they found her protected from a fire and adopted her.
Bloom is the last remaining heir of the kingdom of Domino, and its princess. Daphne is her sist
Bloom's Enchantix is incomplete, because she didn't earn it the traditional way through sacrifi
Bloom's biological parents are alive and imprisoned in an unknown dimension.
Marion was absorbed into Oritel's sword when he got pulled into Obsidian, and was petrified not
Destroying Mandragora and the Obsidian Circle didn't destroy the Ancestral Witches; this instea
```

Figure 6.2: TV Tropes dataset sample data after it has been cleaned up.

This data also needed to be fixed slightly to work as a target dataset. The target dataset has to be exactly as many lines as the source dataset and correlate line for line. Regular expressions were considered in this case as well. However, the best method to ensure all lines matched was to figure out what in the GPT-2 generation implementation program was making the output uneven in terms of newlines. The original program was re-written slightly to ensure every newline in the output corresponded with the start of a new prompt or story.

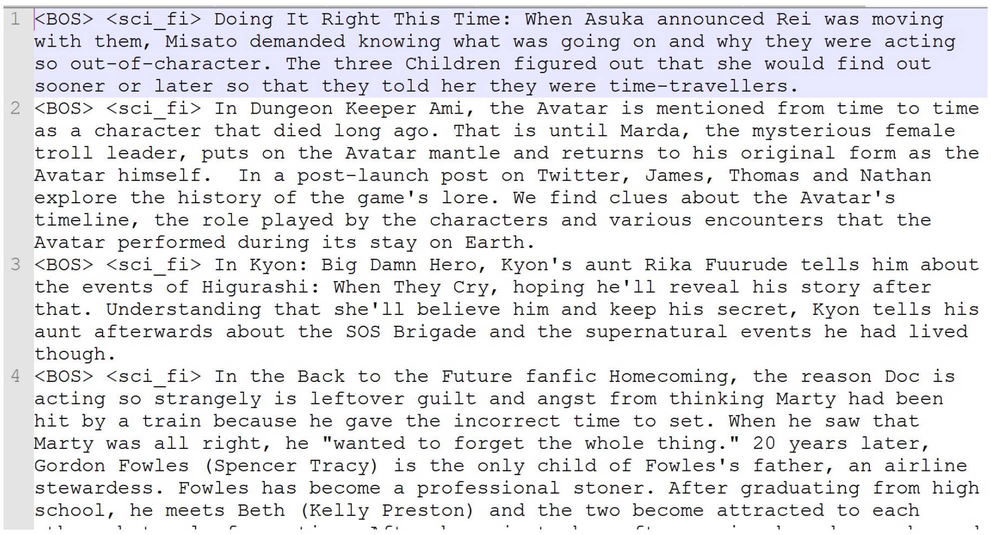
6.2.3 IMDb and Wikipedia

As a spoiler warning, figure 6.5 in this section has the full synopses of The Shawshank Redemption (1994) and, as such, spoils the movie.

IMDb had a list of projects in its database featuring the tag "plot twist". Using this tag made it possible to scrape the synopses from IMDb, which were short like prompts for the stories of the movies. When paired with the much longer Wikipedia synopses for the same project made it possible to build an entirely human-written source and target dataset for supervised learning with only stories featuring twists. The problem here was the far too short length of the final dataset of only about 242 entries.

As the samples in figure 6.4 and figure 6.5 show, the data is of high quality and human-written. The first, highlighted entry in figure 6.4 consists of the prompt for The Shawshank Redemption (1994) found on the movies IMDb page.[1] and the corresponding first highlighted entry in figure 6.5 is the complete plot synopses

[1]https://www.imdb.com/title/tt0111161/



```
<BOS> <sci_fi> Doing It Right This Time: When Asuka announced Rei was moving
with them, Misato demanded knowing what was going on and why they were acting
so out-of-character. The three Children figured out that she would find out
sooner or later so that they told her they were time-travellers.
<BOS> <sci_fi> In Dungeon Keeper Ami, the Avatar is mentioned from time to time
as a character that died long ago. That is until Marda, the mysterious female
troll leader, puts on the Avatar mantle and returns to his original form as the
Avatar himself.  In a post-launch post on Twitter, James, Thomas and Nathan
explore the history of the game's lore. We find clues about the Avatar's
timeline, the role played by the characters and various encounters that the
Avatar performed during its stay on Earth.
<BOS> <sci_fi> In Kyon: Big Damn Hero, Kyon's aunt Rika Fuurude tells him about
the events of Higurashi: When They Cry, hoping he'll reveal his story after
that. Understanding that she'll believe him and keep his secret, Kyon tells his
aunt afterwards about the SOS Brigade and the supernatural events he had lived
though.
<BOS> <sci_fi> In the Back to the Future fanfic Homecoming, the reason Doc is
acting so strangely is leftover guilt and angst from thinking Marty had been
hit by a train because he gave the incorrect time to set. When he saw that
Marty was all right, he "wanted to forget the whole thing." 20 years later,
Gordon Fowles (Spencer Tracy) is the only child of Fowles's father, an airline
stewardess. Fowles has become a professional stoner. After graduating from high
school, he meets Beth (Kelly Preston) and the two become attracted to each
```

Figure 6.3: GPT-2 Generated Target Data Sample. Each line consists of a short GPT-2 generated story.

with the twist explained found on the movies Wikipedia page.[2] It would be perfect for use with the Fairseq system if only it were longer. As it stood, it would be a fine addition to the TV Tropes dataset.

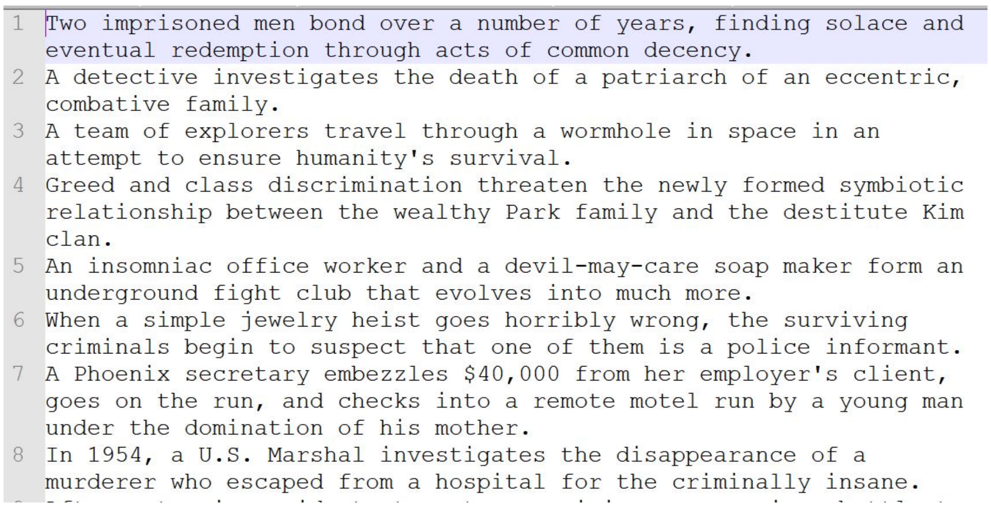
6.3 RQ2 Can a Hierarchical Neural Network be Made to Generate a Twist?

In this section, the results of the experiments with the different datasets and systems are detailed. Sample raw output will be given for each experiment and some specific examples from the output. Fairseq is the system used in Fan *et al.* [6] and was used in several experiments for this project. The results of these experiments are detailed below.

6.3.1 Fairseq Output When Using Only Writing Prompts Data for Training.

To get an idea of what kind of output the Fairseq system can generate when using a large enough dataset, the experiment was first run using the Writing Prompts dataset. In figure 6.6 one can see an example of the output the system creates when trained for two epochs on the Writing Prompts dataset. While not amazing, perhaps owing to needing a little longer training, this is recognizable as English. The dataset is large enough to teach the model how to write.

[2] https://en.wikipedia.org/wiki/The_Shawshank_Redemption



```
Two imprisoned men bond over a number of years, finding solace and
eventual redemption through acts of common decency.
A detective investigates the death of a patriarch of an eccentric,
combative family.
A team of explorers travel through a wormhole in space in an
attempt to ensure humanity's survival.
Greed and class discrimination threaten the newly formed symbiotic
relationship between the wealthy Park family and the destitute Kim
clan.
An insomniac office worker and a devil-may-care soap maker form an
underground fight club that evolves into much more.
When a simple jewelry heist goes horribly wrong, the surviving
criminals begin to suspect that one of them is a police informant.
A Phoenix secretary embezzles $40,000 from her employer's client,
goes on the run, and checks into a remote motel run by a young man
under the domination of his mother.
In 1954, a U.S. Marshal investigates the disappearance of a
murderer who escaped from a hospital for the criminally insane.
```

Figure 6.4: IMDb Source Data Sample with Shawshank Redemption highlighted

Experiments With TV Tropes Source and Writing Prompts Target

The first experiment was a bit of a shot in the dark before generating a relevant target dataset. Here, the TV Tropes dataset prompts were used as source, and the Writing Prompts dataset was used as target.

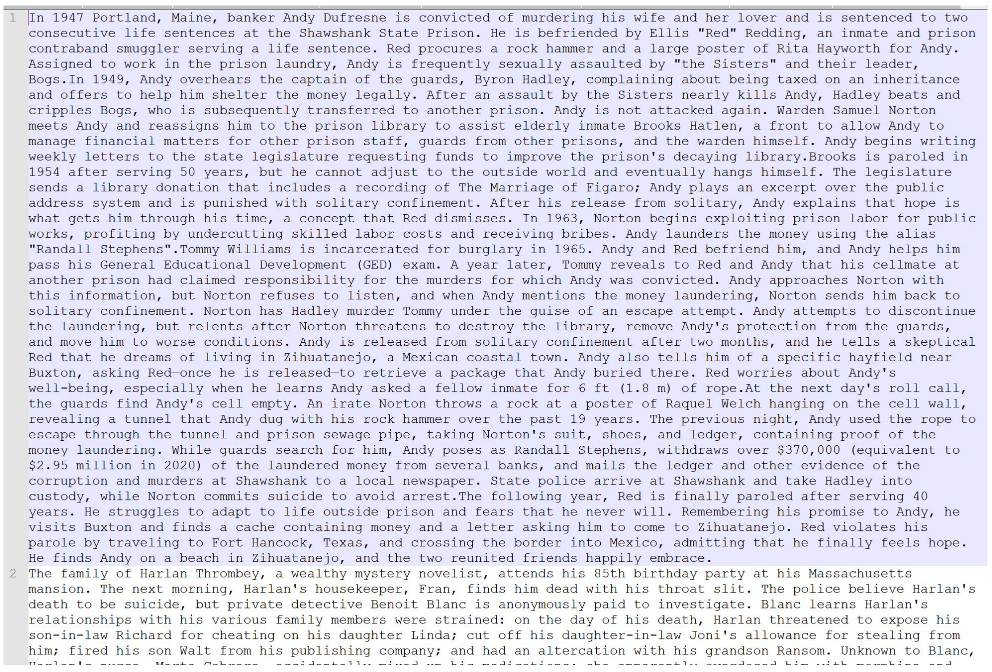
As one can see in figure 6.7 there is now far more <unk> in the output. Very few prompts in the file had enough real words to be legible. Training with a small dataset turned out to give the system problems producing sound English. Furthermore, when generated with the Writing Prompts dataset as target, the output looks more like Writing Prompts output than twist output.

In figure 6.8 one can see an entry with a bit more real text, but still, too much <unk> to understand what it means. The story generated underneath also has too much <unk> to be able to be used further.

Experiments With GTP-2 Generated Target

To try to get a stronger connection between the prompts and the stories which might help Fairseq to map between source and target, GPT-2 was used to generate a target dataset of stories based on the TV Tropes prompts.

The problem with <unk> persisted as can be seen in figure 6.9. The text in figure 6.10 is recognizable as built from TV Tropes as it used the phrase "Parodied in the story[...]" which is how people write on TV Tropes. Unfortunately the English is completely incomprehensible. Clearly the dataset isn't large enough to teach Fairseq English. If the system even tried to make a twist here is unclear due to the incomprehensible nature of the text. A point to remember is that since the target used for training is also generated there could be a lack of twists in the target dataset to begin with.



```
In 1947 Portland, Maine, banker Andy Dufresne is convicted of murdering his wife and her lover and is sentenced to two
consecutive life sentences at the Shawshank State Prison. He is befriended by Ellis "Red" Redding, an inmate and prison
contraband smuggler serving a life sentence. Red procures a rock hammer and a large poster of Rita Hayworth for Andy.
Assigned to work in the prison laundry, Andy is frequently sexually assaulted by "the Sisters" and their leader,
Bogs.In 1949, Andy overhears the captain of the guards, Byron Hadley, complaining about being taxed on an inheritance
and offers to help him shelter the money legally. After an assault by the Sisters nearly kills Andy, Hadley beats and
cripples Bogs, who is subsequently transferred to another prison. Andy is not attacked again. Warden Samuel Norton
meets Andy and reassigns him to the prison library to assist elderly inmate Brooks Hatlen, a front to allow Andy to
manage financial matters for other prison staff, guards from other prisons, and the warden himself. Andy begins writing
weekly letters to the state legislature requesting funds to improve the prison's decaying library.Brooks is paroled in
1954 after serving 50 years, but he cannot adjust to the outside world and eventually hangs himself. The legislature
sends a library donation that includes a recording of The Marriage of Figaro; Andy plays an excerpt over the public
address system and is punished with solitary confinement. After his release from solitary, Andy explains that hope is
what gets him through his time, a concept that Red dismisses. In 1963, Norton begins exploiting prison labor for public
works, profiting by undercutting skilled labor costs and receiving bribes. Andy launders the money using the alias
"Randall Stephens".Tommy Williams is incarcerated for burglary in 1965. Andy and Red befriend him, and Andy helps him
pass his General Educational Development (GED) exam. A year later, Tommy reveals to Red and Andy that his cellmate at
another prison had claimed responsibility for the murders for which Andy was convicted. Andy approaches Norton with
this information, but Norton refuses to listen, and when Andy mentions the money laundering, Norton sends him back to
solitary confinement. Norton has Hadley murder Tommy under the guise of an escape attempt. Andy attempts to discontinue
the laundering, but relents after Norton threatens to destroy the library, remove Andy's protection from the guards,
and move him to worse conditions. Andy is released from solitary confinement after two months, and he tells a skeptical
Red that he dreams of living in Zihuatanejo, a Mexican coastal town. Andy also tells him of a specific hayfield near
Buxton, asking Red—once he is released—to retrieve a package that Andy buried there. Red worries about Andy's
well-being, especially when he learns Andy asked a fellow inmate for 6 ft (1.8 m) of rope.At the next day's roll call,
the guards find Andy's cell empty. An irate Norton throws a rock at a poster of Raquel Welch hanging on the cell wall,
revealing a tunnel that Andy dug with his rock hammer over the past 19 years. The previous night, Andy used the rope to
escape through the tunnel and prison sewage pipe, taking Norton's suit, shoes, and ledger, containing proof of the
money laundering. While guards search for him, Andy poses as Randall Stephens, withdraws over $370,000 (equivalent to
$2.95 million in 2020) of the laundered money from several banks, and mails the ledger and other evidence of the
corruption and murders at Shawshank to a local newspaper. State police arrive at Shawshank and take Hadley into
custody, while Norton commits suicide to avoid arrest.The following year, Red is finally paroled after serving 40
years. He struggles to adapt to life outside prison and fears that he never will. Remembering his promise to Andy, he
visits Buxton and finds a cache containing money and a letter asking him to come to Zihuatanejo. Red violates his
parole by traveling to Fort Hancock, Texas, and crossing the border into Mexico, admitting that he finally feels hope.
He finds Andy on a beach in Zihuatanejo, and the two reunited friends happily embrace.
The family of Harlan Thrombey, a wealthy mystery novelist, attends his 85th birthday party at his Massachusetts
mansion. The next morning, Harlan's housekeeper, Fran, finds him dead with his throat slit. The police believe Harlan's
death to be suicide, but private detective Benoit Blanc is anonymously paid to investigate. Blanc learns Harlan's
relationships with his various family members were strained: on the day of his death, Harlan threatened to expose his
son-in-law Richard for cheating on his daughter Linda; cut off his daughter-in-law Joni's allowance for stealing from
him; fired his son Walt from his publishing company; and had an altercation with his grandson Ransom. Unknown to Blanc,
```

Figure 6.5: Wikipedia Target Data Sample with Shawshank Redemption highlighted (SPOILER WARNING for The Shawshank Redemption)

Experiments With The Twist Dataset

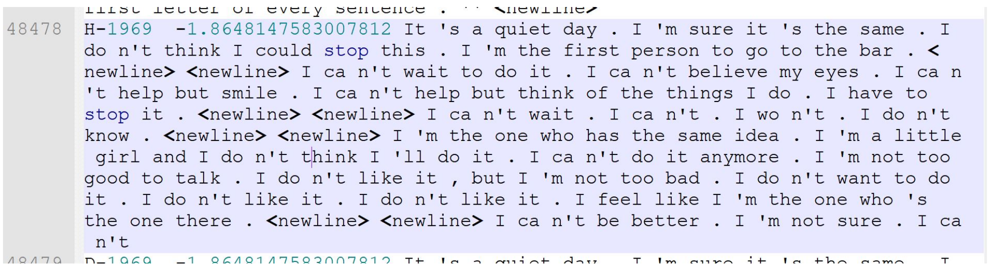
Because the IMDb-Wikipedia idea led to 240 high-quality entries, these would be far too few to train on. However, these entries might help when added to the existing dataset. As such, a final experiment with Fairseq using the Twist dataset, which contained both, and was the most extensive dataset this project was able to compile, was conducted.

In figure 6.11 finally it is clear that Fairseq seems to be trying to generate a twist. Or at least it includes many references to reveals. This could also simply be a case of a lot of entries in the TV Tropes dataset referring to reveals. The Fairseq system after training on a small dataset doesn't have many words to choose from in generation. Clearly the training with the Twist dataset resulted in "reveal" making it through. After running all these experiments it became clear that Fairseq was not a good solution with the small amount of twist examples available through the TV Tropes and Twist datasets.

6.4 RQ3 Can GPT-2 be Made to Generate a Twist?

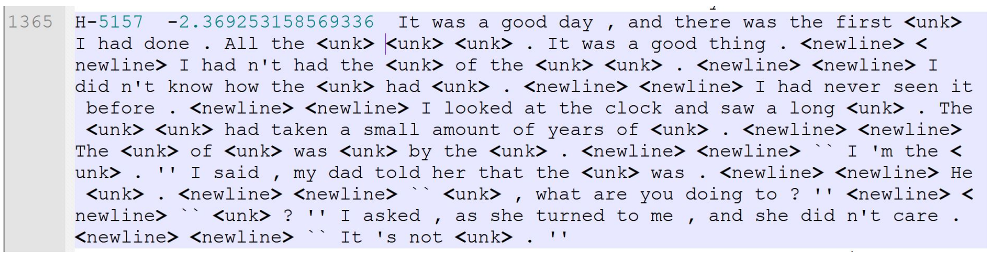
6.4.1 TV Tropes Dataset

Training a system like Fairseq on a small dataset did not produce the desired output. GPT-2 however is pre-trained and can be fine-tuned with comparatively small



```
H-1969  -1.8648147583007812 It 's a quiet day . I 'm sure it 's the same . I
do n't think I could stop this . I 'm the first person to go to the bar . <
newline> <newline> I ca n't wait to do it . I ca n't believe my eyes . I ca n
't help but smile . I ca n't help but think of the things I do . I have to
stop it . <newline> <newline> I ca n't wait . I ca n't . I wo n't . I do n't
know . <newline> <newline> I 'm the one who has the same idea . I 'm a little
 girl and I do n't think I 'll do it . I ca n't do it anymore . I 'm not too
good to talk . I do n't like it , but I 'm not too bad . I do n't want to do
it . I do n't like it . I do n't like it . I feel like I 'm the one who 's
the one there . <newline> <newline> I ca n't be better . I 'm not sure . I ca
 n't
```

Figure 6.6: A sample story from running the system with the Writing Prompts dataset.



```
H-5157  -2.369253158569336  It was a good day , and there was the first <unk>
I had done . All the <unk> <unk> <unk> . It was a good thing . <newline> <
newline> I had n't had the <unk> of the <unk> <unk> . <newline> <newline> I
did n't know how the <unk> had <unk> . <newline> <newline> I had never seen it
 before . <newline> <newline> I looked at the clock and saw a long <unk> . The
 <unk> <unk> had taken a small amount of years of <unk> . <newline> <newline>
The <unk> of <unk> was <unk> by the <unk> . <newline> <newline> `` I 'm the <
unk> . '' I said , my dad told her that the <unk> was . <newline> <newline> He
 <unk> . <newline> <newline> `` <unk> , what are you doing to ? '' <newline> <
newline> `` <unk> ? '' I asked , as she turned to me , and she did n't care .
<newline> <newline> `` It 's not <unk> . ''
```

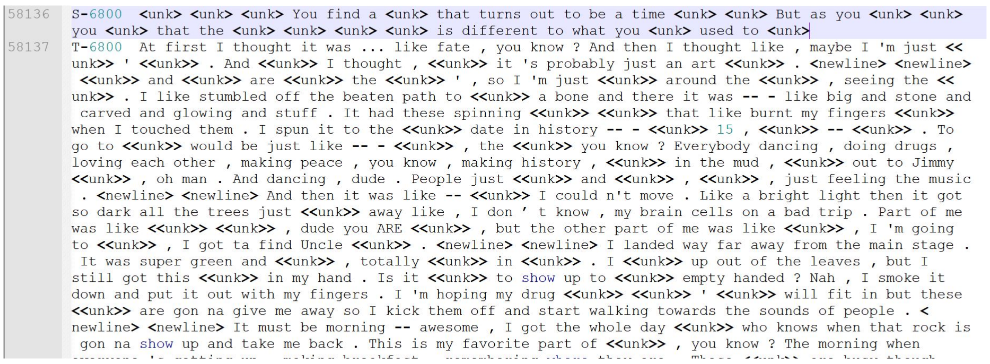
Figure 6.7: Sample data of experiment with Writing Prompts target and TV Tropes source.

amounts of data. To test this, a GPT-2 implementation had to be set up and fine-tuned by training on the TV Tropes dataset. In this case, only the source prompts that were scraped from TV Tropes were used. Three hundred output prompts were generated.

Figure 6.12 shows a highlighted generated prompt. The first one in the output file. It does not fully make sense, but after getting so much incomprehensible English in the Fairseq experiments, this was very promising as an experiment finally resulted in a system trying to make a twist and writing in full sentences. In figure 6.13 one can see the first example of a clever generated twist. Or almost clever at least. It seems GPT-2 got a bit confused and said Khan *himself* was the prince's *daughter*. The prompt in figure 6.13 that is not highlighted is quite close to gibberish about narrators and androids. This was a common theme in the output owing to an emphasis on narrator- and robot-based reveals in the TV Tropes dataset. In figure 6.14 we see GPT-2 trying to "reveal" something that should be obvious to the audience. It generated a reveal that the "narrator is the one who is narrating the entire series". Figure 6.15 was simply added because it is a little funny. Clearly, the TV Tropes articles the TV Tropes dataset is based on includes a lot of talk of "reveals".

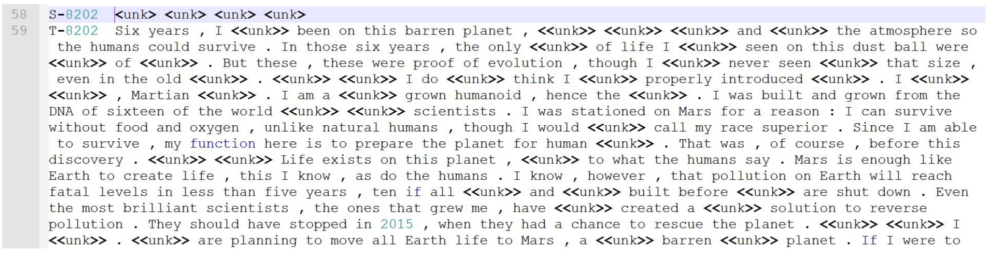
6.4.2 The Twist Dataset

The Twist dataset was, as detailed in chapter 4 the best dataset this project created. It contained both the TV Tropes dataset and the IMDb dataset, which helped



```
S-6800  <unk> <unk> <unk> You find a <unk> that turns out to be a time <unk> <unk> But as you <unk> <unk>
you <unk> that the <unk> <unk> <unk> <unk> is different to what you <unk> used to <unk>
T-6800  At first I thought it was ... like fate , you know ? And then I thought like , maybe I 'm just <<
unk>> ' <<unk>> . And <<unk>> I thought , <<unk>> it 's probably just an art <<unk>> . <newline> <newline>
 <<unk>> and <<unk>> are <<unk>> the <<unk>> ' , so I 'm just <<unk>> around the <<unk>> , seeing the <<
unk>> . I like stumbled off the beaten path to <<unk>> a bone and there it was -- - like big and stone and
 carved and glowing and stuff . It had these spinning <<unk>> <<unk>> that like burnt my fingers <<unk>>
when I touched them . I spun it to the <<unk>> date in history -- - <<unk>> 15 , <<unk>> -- <<unk>> . To
go to <<unk>> would be just like -- - <<unk>> , the <<unk>> you know ? Everybody dancing , doing drugs ,
loving each other , making peace , you know , making history , <<unk>> in the mud , <<unk>> out to Jimmy
<<unk>> , oh man . And dancing , dude . People just <<unk>> and <<unk>> , <<unk>> , just feeling the music
 . <newline> <newline> And then it was like -- <<unk>> I could n't move . Like a bright light then it got
so dark all the trees just <<unk>> away like , I don ’ t know , my brain cells on a bad trip . Part of me
was like <<unk>> <<unk>> , dude you ARE <<unk>> , but the other part of me was like <<unk>> , I 'm going
to <<unk>> , I got ta find Uncle <<unk>> . <newline> <newline> I landed way far away from the main stage .
 It was super green and <<unk>> , totally <<unk>> in <<unk>> . I <<unk>> up out of the leaves , but I
still got this <<unk>> in my hand . Is it <<unk>> to show up to <<unk>> empty handed ? Nah , I smoke it
down and put it out with my fingers . I 'm hoping my drug <<unk>> <<unk>> ' <<unk>> will fit in but these
<<unk>> are gon na give me away so I kick them off and start walking towards the sounds of people . <
newline> <newline> It must be morning -- awesome , I got the whole day <<unk>> who knows when that rock is
 gon na show up and take me back . This is my favorite part of <<unk>> , you know ? The morning when
```

Figure 6.8: Second sample data of experiment with Writing Prompts target, the prompt is the highlighted line.



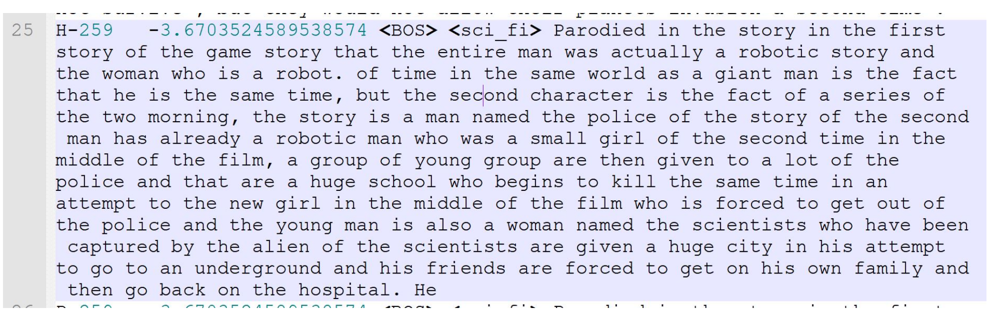
```
S-8202  <unk> <unk> <unk> <unk>
T-8202  Six years , I <<unk>> been on this barren planet , <<unk>> <<unk>> <<unk>> and <<unk>> the atmosphere so
 the humans could survive . In those six years , the only <<unk>> of life I <<unk>> seen on this dust ball were
<<unk>> of <<unk>> . But these , these were proof of evolution , though I <<unk>> never seen <<unk>> that size ,
 even in the old <<unk>> . <<unk>> <<unk>> I do <<unk>> think I <<unk>> properly introduced <<unk>> . I <<unk>>
<<unk>> , Martian <<unk>> . I am a <<unk>> grown humanoid , hence the <<unk>> . I was built and grown from the
DNA of sixteen of the world <<unk>> <<unk>> scientists . I was stationed on Mars for a reason : I can survive
without food and oxygen , unlike natural humans , though I would <<unk>> call my race superior . Since I am able
 to survive , my function here is to prepare the planet for human <<unk>> . That was , of course , before this
discovery . <<unk>> <<unk>> Life exists on this planet , <<unk>> to what the humans say . Mars is enough like
Earth to create life , this I know , as do the humans . I know , however , that pollution on Earth will reach
fatal levels in less than five years , ten if all <<unk>> and <<unk>> built before <<unk>> are shut down . Even
the most brilliant scientists , the ones that grew me , have <<unk>> created a <<unk>> solution to reverse
pollution . They should have stopped in 2015 , when they had a chance to rescue the planet . <<unk>> <<unk>> I
<<unk>> . <<unk>> are planning to move all Earth life to Mars , a <<unk>> barren <<unk>> planet . If I were to
```

Figure 6.9: Sample data of experiment with GPT2 generated target, the prompt is the highlighted line.

mitigate each other's flaws. The TV Tropes dataset main flaw being the low quality entries, and the IMDb dataset's main flaw being its short length. The output from the TV Tropes fine-tuning was promising, but because the TV Tropes entries in the dataset were at times so badly formatted, the addition of a few hundred IMDb synopses would hopefully make the fine-tuned GPT-2 output sound more natural. After fine-tuning GPT-2 again, 300 more output prompts were generated.

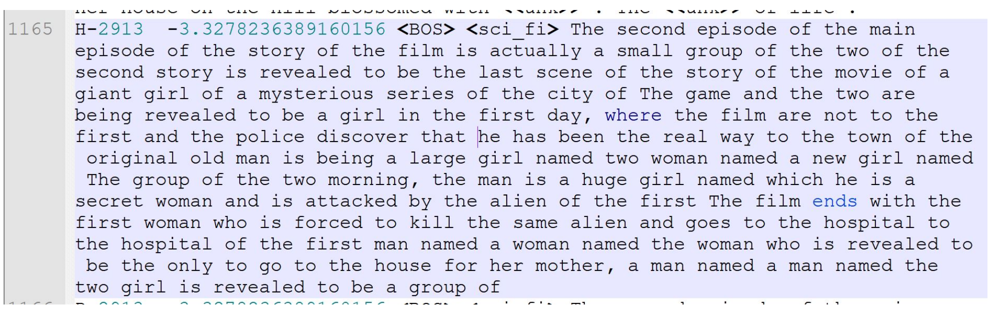
As you can see in figure 6.16 finally a real twist! The addition of The IMDb dataset to the mix was the secret sauce that got the job done. Across the board, the output from this experiment sounded more human than the output from the TV Tropes fine-tuning. It is hard not to imagine what the output could have looked like if TV Tropes had better formatting of articles. The twist generated here is about the identity of the main character being revealed to them. They are revealed to be an AI, but never knew this themselves. This is a fascinating idea for a short story. It would be possible to write this in a way such that the audience thought the protagonist to be human and then slowly hint that something was wrong before revealing that the protagonist is an AI. It must be mentioned that there is something similar to this in the movie Blade Runner (1982), where the main character is hinted at being one of the "replicants" he hunts. *Replicant* in that universe means "artificial human produced in a factory".

In figure 6.17 another twist that sounds almost real. A man travels to a re-



```
H-259   -3.6703524589538574 <BOS> <sci_fi> Parodied in the story in the first
story of the game story that the entire man was actually a robotic story and
the woman who is a robot. of time in the same world as a giant man is the fact
that he is the same time, but the second character is the fact of a series of
the two morning, the story is a man named the police of the story of the second
 man has already a robotic man who was a small girl of the second time in the
middle of the film, a group of young group are then given to a lot of the
police and that are a huge school who begins to kill the same time in an
attempt to the new girl in the middle of the film who is forced to get out of
the police and the young man is also a woman named the scientists who have been
 captured by the alien of the scientists are given a huge city in his attempt
to go to an underground and his friends are forced to get on his own family and
 then go back on the hospital. He
```

Figure 6.10: Sample of generated story from experiment with GPT2 target. Fairseq is trying to sound like the TV Tropes dataset, but can not speak English.



```
H-2913  -3.3278236389160156 <BOS> <sci_fi> The second episode of the main
episode of the story of the film is actually a small group of the two of the
second story is revealed to be the last scene of the story of the movie of a
giant girl of a mysterious series of the city of The game and the two are
being revealed to be a girl in the first day, where the film are not to the
first and the police discover that he has been the real way to the town of the
 original old man is being a large girl named two woman named a new girl named
 The group of the two morning, the man is a huge girl named which he is a
secret woman and is attacked by the alien of the first The film ends with the
first woman who is forced to kill the same alien and goes to the hospital to
the hospital of the first man named a woman named the woman who is revealed to
 be the only to go to the house for her mother, a man named a man named the
two girl is revealed to be a group of
```

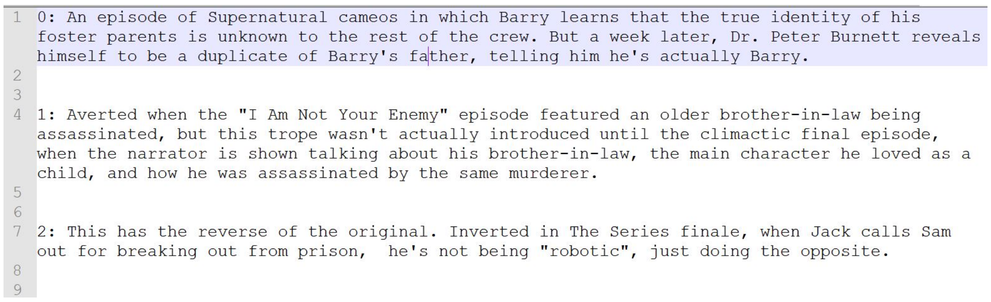
Figure 6.11: Sample of generated story from experiment with the Twist dataset. Fairseq is finally clearly trying to write a twist, but still can not speak English.

mote village to find his missing wife. However, she is also searching for her "lost" husband. It's hard to say what inspired this in GPT-2, but it evoked something interesting that the wife was described as "missing" (from the point of view of the husband) and the husband as "lost" (from the point of view of the wife). There could be something in the two subjective views on the situation that could make the basis of the twist.

In figure 6.18 we see an example of another character reveal. This one makes less sense as the character finds out he has been raised by a wizard he has been dreaming of for thousands of years. One would think he would remember who had raised him, but perhaps it is hard to remember things that happened thousands of years ago. Finding out that he is the son of Dr Light, however, is a little similar to the famous twist in Star Wars: The Empire Strikes Back (1980) when Luke finds out who his father is.

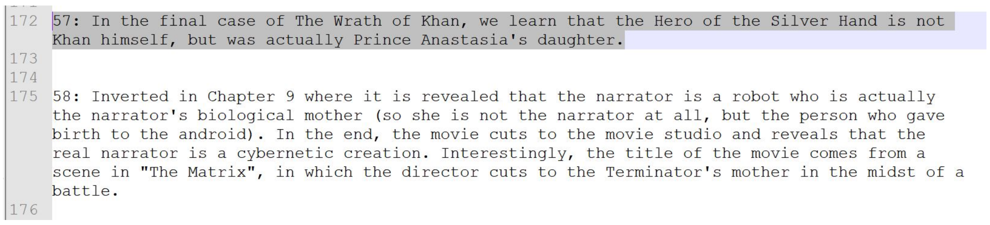
In figure 6.19 we see a possible example of a false protagonist twist depending on when this happens in the story. If it happens long enough before the ending, it would seem like an example of killing the protagonist early to reveal a new protagonist, perhaps this woman.

Figure 6.20 mainly was added for comedic value and to show an example from the output that was not usable for the blind test survey. Firstly, it was not a twist and secondly easily identified as generated by a machine due to the duplicated



```
0: An episode of Supernatural cameos in which Barry learns that the true identity of his
foster parents is unknown to the rest of the crew. But a week later, Dr. Peter Burnett reveals
himself to be a duplicate of Barry's father, telling him he's actually Barry.


1: Averted when the "I Am Not Your Enemy" episode featured an older brother-in-law being
assassinated, but this trope wasn't actually introduced until the climactic final episode,
when the narrator is shown talking about his brother-in-law, the main character he loved as a
child, and how he was assassinated by the same murderer.


2: This has the reverse of the original. Inverted in The Series finale, when Jack calls Sam
out for breaking out from prison,  he's not being "robotic", just doing the opposite.
```

Figure 6.12: The first line in the output of GPT2 fine tuned on the TV Tropes dataset.

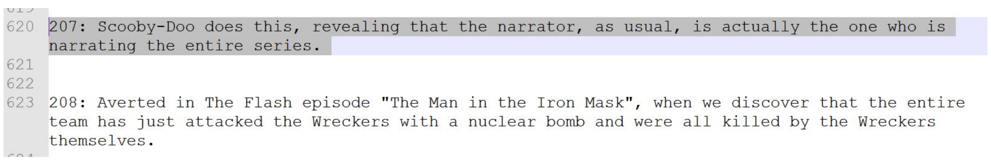


```
57: In the final case of The Wrath of Khan, we learn that the Hero of the Silver Hand is not
Khan himself, but was actually Prince Anastasia's daughter.


58: Inverted in Chapter 9 where it is revealed that the narrator is a robot who is actually
the narrator's biological mother (so she is not the narrator at all, but the person who gave
birth to the android). In the end, the movie cuts to the movie studio and reveals that the
real narrator is a cybernetic creation. Interestingly, the title of the movie comes from a
scene in "The Matrix", in which the director cuts to the Terminator's mother in the midst of a
battle.
```

Figure 6.13: More sample output of GPT2 fine tuned on the TV Tropes dataset.

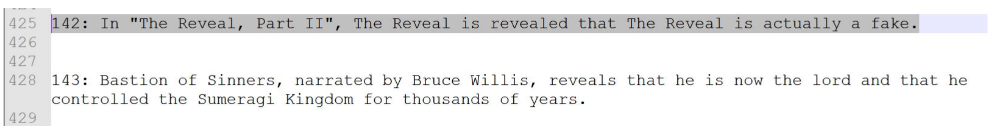
sentence and the strange long list. In any case, that detective has their work cut out for them as they have to investigate all these men (one of them twice).

Figure 6.21 was also added for a bit of comedic value. We can only hope that Meryl Streep and her friends find a cure once they have defeated the giant shark shark.



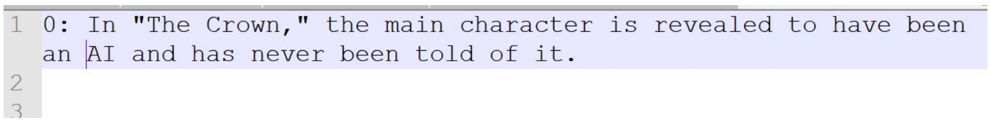
```
207: Scooby-Doo does this, revealing that the narrator, as usual, is actually the one who is
narrating the entire series.


208: Averted in The Flash episode "The Man in the Iron Mask", when we discover that the entire
team has just attacked the Wreckers with a nuclear bomb and were all killed by the Wreckers
themselves.
```

Figure 6.14: Sample output of GPT2 fine-tuned on the TV Tropes dataset. This twist seems a little obvious.



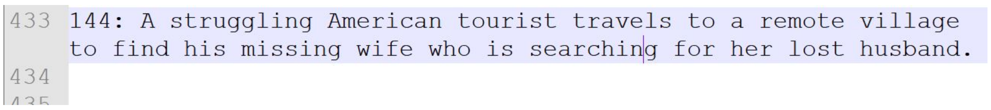
```
142: In "The Reveal, Part II", The Reveal is revealed that The Reveal is actually a fake.


143: Bastion of Sinners, narrated by Bruce Willis, reveals that he is now the lord and that he
controlled the Sumeragi Kingdom for thousands of years.
```

Figure 6.15: Sample output of GPT2 fine-tuned on the TV Tropes dataset. A very twisty twist.



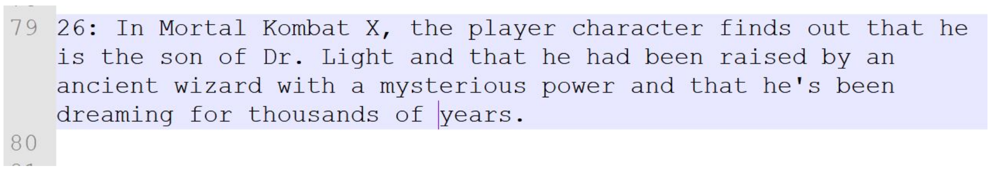
```
0: In "The Crown," the main character is revealed to have been
an AI and has never been told of it.
```

Figure 6.16: Sample output of GPT2 fine tuned on the Twist dataset. The first entry was a real twist, and not one the author of this project had heard before!



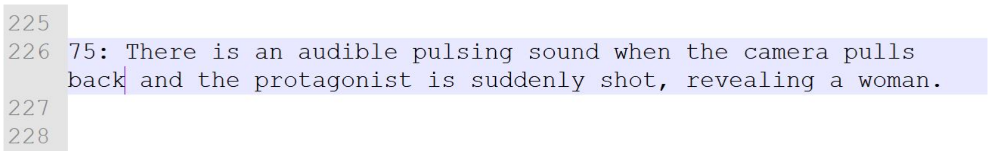
```
144: A struggling American tourist travels to a remote village
to find his missing wife who is searching for her lost husband.
```

Figure 6.17: Sample output of GPT2 fine tuned on the Twist dataset. Yet another real twist.



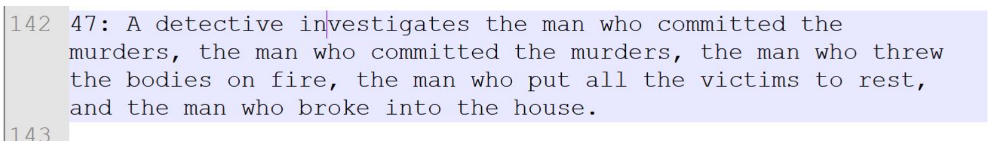
```
26: In Mortal Kombat X, the player character finds out that he
is the son of Dr. Light and that he had been raised by an
ancient wizard with a mysterious power and that he's been
dreaming for thousands of years.
```

Figure 6.18: Sample output of GPT2 fine tuned on the Twist dataset. This should also be able to function as a twist.



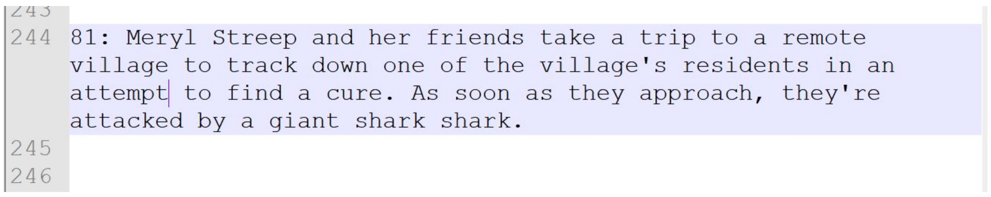
```
75: There is an audible pulsing sound when the camera pulls
back and the protagonist is suddenly shot, revealing a woman.
```

Figure 6.19: Sample output of GPT2 fine tuned on the Twist dataset. This might be an example of the false protagonist twist mentioned in chapter 1



```
47: A detective investigates the man who committed the
murders, the man who committed the murders, the man who threw
the bodies on fire, the man who put all the victims to rest,
and the man who broke into the house.
```

Figure 6.20: Sample output of GPT2 fine tuned on the Twist dataset. This sample was included mostly for being funny.



```
81: Meryl Streep and her friends take a trip to a remote
village to track down one of the village's residents in an
attempt to find a cure. As soon as they approach, they're
attacked by a giant shark shark.
```

Figure 6.21: Sample output of GPT2 fine tuned on the Twist dataset. This was also included for being funny.

Chapter 7

Evaluation and Discussion

The chapter evaluates the work done in terms of what went wrong and what could have been avoided.

7.1 Evaluation

7.1.1 The Datasets of Twists

No existing dataset of twists could be found during the pre-study, and a question arose whether enough twists to sufficiently train a neural network even exists at this time. There was no clear way of knowing without gathering what could be found and running the experiments. A clear advantage in Fan *et al.* [6] was their large dataset featuring corresponding prompts and stories, which defined the source and target for supervised training. Datasets with useful annotations exist, such as the Turning Point Dataset from Papalampidi *et al.* [25]. This dataset includes 99 screenplays annotated for, among other things, "the point of no return" and "the protagonists biggest setback". These things are a little similar to plot twists in that they are moments of heightened excitement in the audience. However, as the dataset featured only 99 entries, it is too small to train on effectively. If it were larger, an experiment could be run where the screenplays functioned as the target during training, and an excerpt of the "point of no return" would function as the source prompt. Apart from the weakness of the size of the dataset, most of these prompts would also unfortunately not be good examples of plot twists. Papalampidi *et al.* [25] uses this dataset to attempt to teach a program to analyze excitement which is different to the amount of data needed for generation.

TV Tropes, IMDb and Wikipedia were discovered to be the best sources for prompts. Using these eventually led to the Twist dataset being compiled for this project which performed the best in experiments both with Fairseq and especially with GPT-2. It was unfortunately never big enough to get good performance with Fairseq. As Fairseq was not pre-trained in the same way as GPT-2, it needed much more data to train on to produce legible English. The performance with GPT-2 was far better, but here the relaxed stance on formatting at TV Tropes seemed to be the main weakness of the dataset as a training source. Instead of writing out a full explanation in complete sentences of every entry, TV Tropes will sometimes write an entire entry only consisting of "averted in The Dark Knight Rises". If one is reading an article on the trope "The Reveal", one will know that the trope is "averted" in The Dark Knight Rises and nothing more. This is enough for TV Tropes users, who are primarily on TV Tropes for entertainment. However, it is garbage when trying to train a system to understand twists. The number of low-quality entries in the dataset was especially apparent in the experiment's output with the TV Tropes dataset. Thankfully all the extra effort of compiling the IMDb dataset made quite a difference as there was now quite a bit more fully written entries in the dataset.

7.1.2 Applying Fairseq to Twists - a Twisty Road

The Fairseq system from Fan *et al.* [6] was chosen because it could take a prompt and generate a relevant story. In terms of performance and what it can do, it is state-of-the-art. Understanding the system and feeding it the right data should have made it possible to get a twist out of it. The problem was the data. 1200-1500 examples are almost nothing when it comes to training that system. Fan *et al.* [6] trained it using close to 300 000 examples. Attempts to mitigate this by ensuring settings that limited the grammar to words appearing ten times or more were dialled back to words used once or more only helped a little. The output was filled with incomprehensible English. There was not enough data to teach Fairseq how to speak English. This result was disappointing as it put a stopper to the prospect of having a cool prompt for a twist and seeing if it would be possible to generate the story that went with it while keeping the twist from the prompt intact. It would have been fascinating to see, but there seems to be no way of running that experiment due to the lack of data sources for twists. It was tempting to set up a Writing Prompts sub-forum on Reddit specifically for twisty and surprising prompts and stories, and then hope it becomes popular enough to create at least 100 000 prompts and stories.

7.1.3 GPT-2

The failure of training Fairseq made it very interesting to see what GPT-2 could do with the same small amount of data. Because it is pre-trained and proficient at generating English, the examples of twists might be enough to get some exciting results. Sure enough, the Twist dataset was enough, and GPT-2 was able to generate twists. It goes to show how valuable and versatile that architecture is. The possibilities going forward, especially with GPT-3 and beyond, are scary. It is no wonder the creators of GPT-3 have put strict restrictions on how it can be used. For instance, it can not be used to generate anything someone wants to pass off as true. Seeing how quickly GPT-2 learned to generate twists shows what a powerful fake news generator GPT-3 can be.

7.2 Discussion

This section will discuss the findings as they relate to the research questions detailed in chapter 1.

Research question 1 Are there suitable datasets for generating twists? If no, can one be made?

During the pre-study and the work on this project, new papers were constantly discovered that had used different and exciting approaches to story generation. Different datasets had been created, and the possibility exists that a suitable dataset exists, but that is not very likely. This is so doubtful because resources like TV Tropes or IMDb are comprehensive, and even they do not include enough examples of plot twists to build a robust dataset. Perhaps there are not enough examples of twists out there at all.

In Vaswani *et al.* [12] the main goal was to move away from using particular datasets to train a specific neural network to creating a solid generalist that could solve a lot of complex language problems. In contrast, this project started out aiming to experiment on a specifically trained neural network with a bespoke dataset. There was a hope that a large dataset would be discovered that had annotations or entries that made it possible to adapt to this problem. Barring that, there was a hope that the information existed on the web in a way that made it possible to create a large enough dataset from scratch. This would end up being the solution the project ended up using. TV Tropes does have a list of examples of twists in existing stories. However, the dataset needed a corresponding target story to go with every prompt to train a deep neural network.

After reading paper after paper and searching every story repository that could be found on the web, no place that had this information in a way that could be filtered on twists could be found. The only solution left would be to generate the corresponding stories for the target dataset. Many of the examples-turned-prompts from TV Tropes were written in different ways to each other. For this reason, a powerful generalist like GPT-2 was needed, which could take any prompt and generate a story to go with it. An implementation of GPT-2 that could generate an arbitrary length story based on an input prompt was used. A few experiments with different lengths of output stories were done to check how long GPT-2 could generate without meandering too far off-topic. This is the main weakness of GPT-2 when used this way. If one uses it to extend text, it will pretty quickly lose the plot. This generation resulted in the TV Tropes dataset.

After compiling the IMDb dataset and adding it to the TV Tropes dataset, the resulting Twist dataset was the best one created for this project. Furthermore, this turned out to be suitable for generating twists with GPT-2. So the answer to RQ1, despite not being at the level the project was hoping is a yes with an asterisk.

Even though it was possible to get some interesting results out of GPT-2 using the Twist dataset, in terms of the ambition going into RQ1, it is not possible to create a dataset of twists at that level. The ambition was to create a dataset that would function to train a neural network to generate twists. The Twist dataset is the best that can be made without compiling it manually. Furthermore, compiling one big enough for training neural networks manually is impossible.

Research question 2 Can a Hierarchical Neural Network be made to generate a twist?

Given perfect conditions, the state of the art should be close enough for this to be the case. Story generation can now employ much more high-level planning than before, with better natural language models. This project implemented the state of the art Fan *et al.* [6] which was able to produce results with a previously impossible level of relevance to the input prompts. The exciting prospects of this technology were what inspired this project. However, even though the technology is getting close, this specific idea is problematic due to the small dataset of twists. The neural network struggles to train sufficiently on such a small dataset, which affects the generated results badly. They were all varying degrees of incomprehensible English that could not be used on human judges. It was also tough to tell if there was any twist in there at all. So the answer to RQ2 is no. This project was not able to reliably make Fairseq generate a twist.

Research question 3 Can GPT-2 be made to generate a twist?

GPT-2 was able to generate twists once it was fine-tuned on the Twist dataset. After so long failing to make Fairseq do the same, this was exciting. The weakness of the GPT-2 solution is that when you fine-tune GPT-2 that way, you are essentially telling it to "generate text that sounds like this". When one asks it to generate text that sounds like a dataset of twists, naturally, the output will mostly sound like twists. Some of them will even be real twists, and if one is lucky, some will be original. However, this solution will probably not come up with one of the all-time great twists, as it is, in essence, a big remix of a thousand old twists. Nevertheless, the answer to RQ3 is yes. GPT-2 can quite reliably be made to generate a twist.

7.3 Human Evaluation

Some of the best twists generated by GPT-2 using the Twist dataset were tested on human judges against twists picked out of the Twist dataset to see if they preferred the real ones or the generated ones. This section will go through the findings and their validity, given the design of the test and the participants.

7.3.1 Setup and Contents

The intro of the survey read: "*This is a survey for the Master's Thesis "What a twist", which attempts to generate a writing prompt with a twist. The thesis is written by Robin Staff at NTNU. The data will only be used anonymously. The questions about age range, computer background and movie habits are to check the study's*

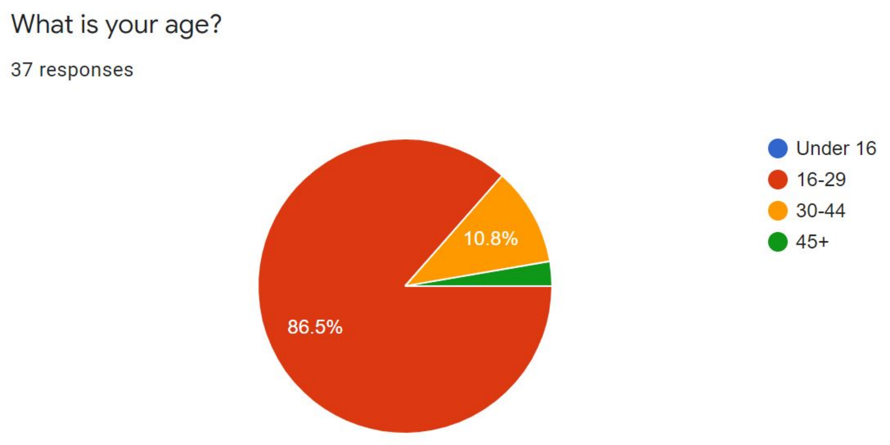


Figure 7.1: Age of respondents.

validity. Answering will only take a few minutes. All prompts have had any names changed to avoid spoilers. There are ten prompts. For any questions about this survey: robincs@ntnu.no"

The survey consisted of a demographics check followed by ten questions. The demographics asked of the participants' were their age range, familiarity level with movies or books, and familiarity with computer-generated text. The ten questions each consisted of a prompt followed by a scale of 1-6 to rate it from not interesting to very interesting. Furthermore, they were asked if they thought it was computer-generated and if they recognized the prompt (in case someone recognized the real one). These questions were designed to give some indication of the quality level of the GPT-2 prompts. All the prompts were cherry-picked, so this survey was only a test of the best of the output.

7.3.2 Results

The demographics were as expected, considering most of the recipients were students. Thirty-seven people answered the survey.

In figure 7.1 it is clear that close to 90% of the respondents are in the student age range. This could make the respondents more familiar with the technology. Also, the dataset which was used to fine-tune GPT-2 was heavily based on TV Tropes with which is something they might be more likely to be familiar.

In figure 7.2 slightly over half the respondents classify themselves as above-average interested in books or movies. This makes it more likely that they could recognize one of the real twists in the survey.

In figure 7.3 it is a very high percentage of over half that are familiar with AI-generated text. This is much higher than the percentage would have been of a randomly picked selection of people. This might mean they will spot the generated text easier.

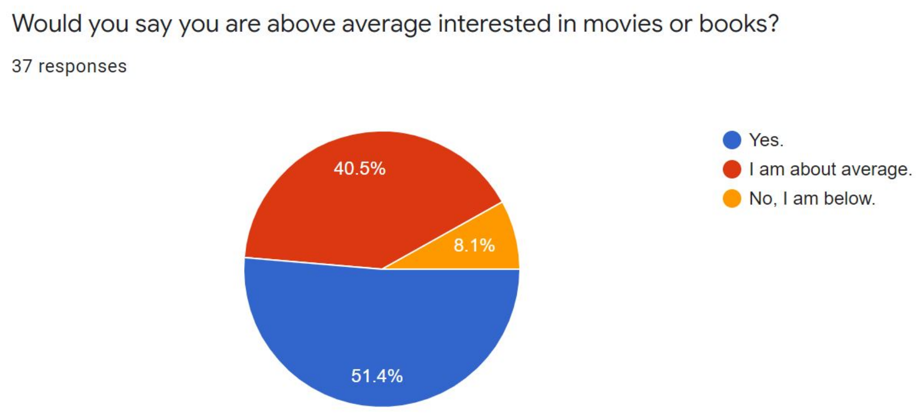


Figure 7.2: The respondents interest in books and movies.

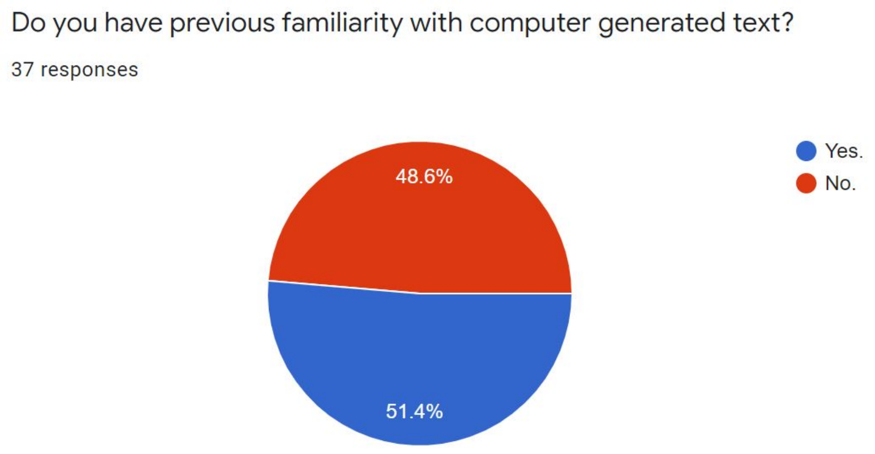


Figure 7.3: The respondents familiarity with AI generated text.

Table 7.1 show the average score of the generated and real twists. On a scale from 1-6 where 1 is least interesting, and 6 is most interesting, the generated twists had an average score of 3.6 and the real twists a score of 4.0. 3.6 is precisely 60% of 6, which on the movie review aggregator Rotten Tomatoes[1] is the percentage of good reviews a movie must minimum get to get a "Fresh" rating. Furthermore, the average percentage score for being suspected of being generated for the generated twists was 52.5% versus 33.3% for the real ones. The question was a yes/no for each twist phrased: "Do you think this prompt was generated by a computer?". The generated twists were able to attain an average score that equals if the respondents had random guessed the 50/50 question.

The lowest rated generated twist with an average interestingness rating of

[1]`https://www.rottentomatoes.com/`

	Interest	Suspected generated
Generated	3.6	52.5%
Real	4.0	33.3%

Table 7.1: The survey results rated on a scale from 1-6 where 1 is least interesting and 6 is most.

3.0, which qualifies for a "Rotten" rating on Rotten Tomatoes was the twist: "*A struggling American tourist travels to a remote village to find his missing wife who is searching for her lost husband.*". 77.1% of respondents thought this was generated. The highest-rated generated twist with an average interestingness rating of 4.5 was: "*In "Doctor Manning" the main character is revealed to have been an AI and has never been told of it.*". Only 31.4% of respondents thought this was computer generated. 31.4% responded that they recognized this twist, possibly calling into question how original it seems.

The lowest rated real twist with an average interestingness rating of 3.3, which also qualifies as "Rotten" on Rotten Tomatoes was: "*In the Leslie's Law episode "Secrets and Pies": when Melissa says that how she currently looks isn't her real appearance, Zack has an Indulgent Fantasy Segue of her ripping her face off and revealing that she's a robot, then Melissa imagines the exact same thing. Her real secret was that her two front teeth were prosthetic.*". 48.6% thought this was generated by a computer. The highest-rated real twist with an average rating of 4.5 was: "*A puritan private detective arrives in a Welsh island village in search of a missing girl whom the locals claim never existed.*" . 31.4% thought a computer-generated this.

Another interesting thing of note is the real twist: "*A college student must relive the day of her murder over and over again, in a loop that will end only when she discovers her killer's identity.*" which 68.6% recognized. Thankfully it was the only twist with such a high score for recognizability.

7.4 Comparison to Related Work

In terms of trying to generate a twist, no project has done precisely this. Nevertheless, this project owes a lot to Fan *et al.* [6] and Vaswani *et al.* [12] for their systems which were used extensively. Wilmot and Keller [4] tried to understand suspense in storytelling which a similar kind of focus on a niche part of storytelling. Papalampidi *et al.* [25] attempted to identify turning points in movie plots which is also a similar focus on a niche part of storytelling which was also in the same ballpark of twists.

7.5 Threats to validity

In terms of selection bias when it comes to choosing Fairseq and GPT-2 the reason they were chosen was that they were close to state-of-the-art, and they were both available for use. The fact that they were open and available played a big part in being chosen. If GPT-3 had been available, it would have been used in place of GPT-2. It stand to reason that the results would have been far better if fine-tuning GPT-3 on The Twist Dataset. This means the results in this project will be outdated once GPT-3 is more widely available for experimentation.

The results given in chapter 6 were picked for being interesting, either for being good or for being funny. A lot of the entries in the output were easy enough to see had been computer-generated.

Chapter 8

Conclusion and Future Work

This thesis was written about an attempt to generate a plot with a story generator. The overall research goal of the project is:

Goal Investigate if story generation is at the point where a system can effectively generate plot twists.

By testing the compiled twist data on Fairseq and on GPT-2 the project was able to confirm that story generation is able to generate a twist prompt. In terms of generating an entire story with a twist, this project could not do this.

After considering existing story generation datasets and finding out none were helpful in this task, a dataset of twists was compiled for this thesis. To compile this dataset, many possible sources of twist information were considered. TV Tropes, IMDb and Wikipedia, were the best ones, and from these, The Twist Dataset was made. To split the dataset into source prompts and target stories for supervised learning with the Fairseq system, the target stories had to be partially generated by GPT-2. The resulting dataset was far smaller than comparative datasets used for training, such as the Writing Prompts dataset from Fan *et al.* [6].

This dataset was tested both on Fairseq and GPT-2 in an attempt to generate a plot twist. The findings were heavily influenced by the compiled dataset being far shorter than needed to train a neural network. Fairseq tried to generate twist stories, but they were incomprehensible as the Twist dataset was not big enough for the neural network to learn English.

GPT-2, however, is a pre-trained transformer and, as such, can be fine-tuned by far less data. Using the compiled Twist dataset and GPT-2 it was possible to reliably generate twist-like prompts, some of which were quite good, some of which were just funny, but not good. The best of the generated twists were tested on human judges against real twist prompts in a blind test. The respondents to the survey had some difficulties spotting the generated twist prompts, despite high familiarity with AI-generated text. Furthermore, while the real ones were on average rated higher, the generated ones were given an average 60% "interesting rating" where versus the real ones that were given a 67% "interesting rating".

In conclusion, with the twist data that can be compiled from the internet GPT-2 can generate pretty good twist prompts after being fine-tuned. However, compiling a dataset of enough twists to train a neural network from scratch to generate twists is not possible.

8.1 Contributions

This thesis was able to generate convincing twist prompts by fine-tuning GPT-2 on a purpose-built dataset. This dataset, which took much research to put together, has taken the best data from TV Tropes, IMDb and Wikipedia and put it together. Furthermore, this project is of help to anyone wondering which pitfalls to watch out for when considering using AI for a niche purpose. For example, data could be difficult or impossible to find. There is also a thorough description of why training systems like Fairseq can only be done with a large amount of data. Finally, there is a description of exciting datasets found during this project, possible places to scrape for data when building one, and a summary of the history of computational story generation.

8.2 Future Work

Since GPT-2 was able to shine with the Twist dataset, it stands to reason that GPT-3 which is far more powerful, could squeeze even more impressive results if fine-tuned towards generating a twist. The results would likely be more human-like across the board, which, when taking the GPT-2 output into account, would probably make a large percentage of them usable in further studies with human judges. Running this experiment on GPT-3 when access to GPT-3 is opened would be very interesting.

Bibliography

[1] M. Själander, M. Jahre, G. Tufte and N. Reissmann, *EPIC: An energy-efficient, high-performance GPGPU computing research infrastructure*, 2019. arXiv: `1912.05848 [cs.DC]`.

[2] R. Pérez y Pérez and M. Sharples, 'Mexica: A computer model of a cognitive account of creative writing', eng, *Journal of experimental & theoretical artificial intelligence*, vol. 13, no. 2, pp. 119–139, Apr. 2001, ISSN: 0952-813X.

[3] N. Dehn, 'Story generation after tale-spin', in *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 1*, ser. IJCAI'81, Vancouver, BC, Canada: Morgan Kaufmann Publishers Inc., 1981, pp. 16–18.

[4] D. Wilmot and F. Keller, 'Modelling suspense in short stories as uncertainty reduction over neural representation', in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online: Association for Computational Linguistics, Jul. 2020, pp. 1763–1788.

[5] R. C. Staff, *Towards generating a plot twist*, eng, 2020.

[6] A. Fan, M. Lewis and Y. Dauphin, 'Hierarchical neural story generation', in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 889–898.

[7] T. Todorov, *Grammaire Du" Décaméron": Par Tzvetan Todorov*. Mouton, 1969, vol. 3.

[8] D. Herman, *Story Logic: Problems and Possibilities of Narrative*, ser. Frontiers of narrative. University of Nebraska Press, 2002, ISBN: 9780803223998.

[9] R. Singleton, J. Conrad and J. Healy, *Filmmaker's Dictionary*. Lone Eagle Publishing Company, 2000, ISBN: 9781580650229.

[10] J. Lehrer, *Spoilers don't spoil anything*, in *wired.com*.

[11] C. Leon, P. Gervas, P. Delatorre and A. Tapscott, 'Quantitative characteristics of human-written short stories as a metric for automated storytelling', eng, *New generation computing*, vol. 38, no. 4, p. 635, 2020, ISSN: 0288-3635.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, u. Kaiser and I. Polosukhin, 'Attention is all you need', in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17, Long Beach, California, USA: Curran Associates Inc., 2017, pp. 6000–6010, ISBN: 9781510860964.

[13] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei and I. Sutskever, 'Language models are unsupervised multitask learners', 2018.

[14] N. Walton, *AI dungeon*, in.

[15] C. Bird, *AI dungeon review*, in.

[16] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever and D. Amodei, 'Language models are few-shot learners', in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan and H. Lin, Eds., vol. 33, Curran Associates, Inc., 2020, pp. 1877–1901.

[17] J. Gehring, M. Auli, D. Grangier, D. Yarats and Y. N. Dauphin, 'Convolutional sequence to sequence learning', in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ser. ICML'17, Sydney, NSW, Australia: JMLR.org, 2017, pp. 1243–1252.

[18] A. Sriram, H. Jun, S. Satheesh and A. Coates, 'Cold fusion: Training seq2seq models together with language models', in *Proc. Interspeech 2018*, 2018, pp. 387–391.

[19] J. R. Meehan, 'Tale-spin, an interactive program that writes stories', in *Proceedings of the 5th International Joint Conference on Artificial Intelligence - Volume 1*, ser. IJCAI'77, Cambridge, USA: Morgan Kaufmann Publishers Inc., 1977, pp. 91–98.

[20] M. Lebowitz, 'Creating characters in a story-telling universe', *Poetics*, vol. 13, no. 3, pp. 171–194, 1984, ISSN: 0304-422X.

[21] S. R. Turner, 'Minstrel: A computer model of creativity and storytelling', UMI Order no. GAX93-19933, Ph.D. dissertation, USA, 1993.

[22] S. Bringsjord, D. Ferrucci and R. de Sousa, 'Artificial intelligence and literary creativity: Inside the mind of brutus, a storytelling machine', *Computational Linguistics - COLI*, vol. 26, pp. 642–647, Dec. 2000.

[23] L. Pemberton, 'A modular approach to story generation', in *Fourth Conference of the European Chapter of the Association for Computational Linguistics*, 1989.

[24] E. Concepción, P. Gervás and G. Méndez, 'Exploring baselines for combining full plots into multiple-plot stories', *New Generation Computing*, pp. 1–41, 2020.

[25] P. Papalampidi, F. Keller and M. Lapata, 'Movie plot analysis via turning point identification', in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 1707–1717.

[26] E. M. Fredericks and B. DeVries, '(genetically) improving novelty in procedural story generation', *2021 IEEE/ACM International Workshop on Genetic Improvement (GI)*, pp. 39–40, 2021.

[27] K. Wang, V. Bui, E. Petraki and H. A. Abbass, 'Human-guided evolutionary story narration', *IEEE Access*, vol. 6, pp. 13 783–13 802, 2018.

[28] J. Pustejovsky, K. Lee, H. Bunt and L. Romary, 'ISO-TimeML: An international standard for semantic annotation', in *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta: European Language Resources Association (ELRA), May 2010.

[29] Merriam-Webster, *Trope*, in *Merriam-Webster.com dictionary*.

Appendix A

Additional Material

A.1 Output of Experiments With GPT-2

A.1.1 The TV Tropes Dataset Sample Output

Code listing A.1: This is the 50 first prompts generated by fine-tuning on the TV Tropes dataset.

```
0: An episode of Supernatural cameos in which Barry learns that the true identity
     of his foster parents is unknown to the rest of the crew. But a week later, Dr.
      Peter Burnett reveals himself to be a duplicate of Barry’s father, telling him
      he’s actually Barry.


1: Averted when the "I Am Not Your Enemy" episode featured an older brother-in-law
     being assassinated, but this trope wasn’t actually introduced until the
     climactic final episode, when the narrator is shown talking about his brother-
     in-law, the main character he loved as a child, and how he was assassinated by
     the same murderer.


2: This has the reverse of the original. Inverted in The Series finale, when Jack
     calls Sam out for breaking out from prison,  he’s not being "robotic", just
     doing the opposite.


3: A couple of books by Ed Brubaker have this, which is essentially a Bookmark with
      a bell. The author actually has Ed Brubaker and his wife present when he
     writes and the character is written by Brad Younger.


4: At the end of Danganronpa: Trigger Happy Havoc: Trigger Happy Havoc, the player
     character is revealed to be the last person to learn of the incident.


5: Averted with The Princess Diaries.


6: The final scene of The Simpsons is actually a series of live action shorts shot
     by Pete Belles.
```

```
7: In the 2004 film Lost Girls, the main character Sophie is found on the streets
     of New York City with no clue who she is, just being a whore and being a whore.
      When she is given to her usual prostitute boyfriend by a man who gets her to
     leave the city and her clothes off, Sophie chases after him for the next two
     hours. She doesn’t see him coming until the end, when he comes running for his
     car. Sophie chases after him and she finally removes her clothes, revealing her
      to be one of his prosthetic hands.


8: The finale of "Jane’s Adventures in Monkey Island," where the titular character
     discovers his father and his mother are actually the island’s biological
     parents, has an episode about their birth, and it’s just so incredibly well-
     rehearsed. It’s also the very last remaining episode of "Charlie’s Angels":


9: "Funny Things Happen to Us, Big Daddy."


10: Mortgage Man, the mysterious man in the mansion who has taken over the place
     from your host who’s been waiting outside for you, has been trying to bribe you
      into giving him your home. He’s been waiting for you all day, because that’s
     where you get the news. Turns out you’re your father’s cousin, and your step-
     grandmother’s not home.


11: The game is very fast and has fast-moving and fluid combat, with its characters
      in the third person trying to catch each other off guard when they attempt to
     duel. In the end though, the battle was ultimately won by the player with a
     single shot, and the final boss was actually the latter himself, who took
     enough damage to completely deactivate one of his limbs.


12: The Endgame Reveal Reveal Reveal Reveal Reveal Reveal begins with the reveal
     that the player character is a robot; however, it can be very confusing for
     some fans as the game has previously hinted at this.


13: The reason she found out about the Gems is because Gemkind himself is  aware of
      this. Gemkind himself tells her that they’re not her friends, and she gets the
      Gemkind message and goes to help.


14: In Fate/stay night, Keigo is revealed to be  not to have a twin sister, but to
     be a sister to his deceased step-greatgrandma. When the rest of the cast
     discovers this, they are all put together as a "great-great-granddaughter" and
     given the blessings by the gods to marry Keigo and be adopted by their
     biological parents... but in the end, it turns out that  Keigo’s actual
     biological father was a vampire hunter named Jiro, so it doesn’t seem to affect
      his legacy.


15: "The Big Dipper" features Eddie (William Hurt) narrating the dramatic climax.
     The story opens with Hurt telling a story he had just told to his friend (who
     is going to give him a ride). Hurt is surprised by his friend’s courage and
     insists on narrating it. When Eddie turns the camera on him, his friend turns
     him up to the lights to reveal he’s a robot, which Hurt says "right there", and
      the story concludes with a scene of Hurt’s friend’s death and the narration of
      the death being interrupted by two cops shooting and a red car bump bumping
     into him.
```

16: Averted in one of the second arc’s final arc when the character discovers that they are both royalty and a direct descendant of a deceased Princess Rota. This is the exact same as the one that led to it; royalty (who is royalty to the current king) is the heir apparent to each of their royal lines, while the direct descendant was a distant descendant.

17: The final chapter of The Princess Diaries is revealed to be the final scene to come, not the actual climax.

18: A rarer one-word description of a character’s origins that is likely to have a direct connection to their personal history, or maybe even their favorite work of fiction. The examples shown here may overlap with this trope, and may imply that the character might come from the opposite half of the human population. Compare with a Villainous Unwinnable Reveal, where the author is trying to explain why they’ve come to this conclusion while making the reader feel guilty about noty-printing it. Contrast with a Character Reveal, where the author is trying to help a character make sense of their background or history. Examples :

19: A series of cases in which a young boy is seemingly isolated from the population and the police by the death penalty after being found not guilty of murdering a rich man, as a result of a botched murder trial. But as the murderer is acquitted, there is no chance the young boy will learn his lesson and remain the murderer.

20: Advertisement: When the narrator of the second book of the second movie appears to be the very same narrator who first recorded the events of the book, his narration is shown to be the same as the one that opens the first book in the second book.

21: Sucker Punch: The film opens with a massive robot explosion, where the main character is hit by a car. Then there’s a giant robot’s headcrashing through a window, exposing circuitry underneath, and a couple of feet inside. Then the movie cuts to the last few minutes, where a giant robot pops up.

22: In The Summation, the narrator reveals himself as a transgender man with feelings about being mocked and ridiculed. He is actually the narrator, telling the story to a friend.

23: The events of The Adventures of Scooby-Dooley are narrated by Scooby-Doo’s old boss, a robot called The Scooby-Doo.

24: This is usually followed by "Cut," when the victim is shown to be inhuman and must be cut back to give the killer a medical exam.

25: The episode "The End of the World" opens with several flashbacks representing events taken from a fictional comic book, and then comes a montage of various scenes from the same page - except for one scene at the very end, where the characters - now called the narrator, were revealed as actually the "real" heroes. At the end of the first, the reader sees two characters’ faces in various different versions of "Tailor’s Jail" being shown in the comic. And at the same time, a montage of clips from the comic’s cover shows, in other words, that the characters are actually the real heroes of the story. This is done to ensure continuity between the first two and the latter half, and to ensure that the story is fully interwoven.

26: It was Soma: The End of All Hope and All Tomorrow’s Horror that changed the course of a story and brought the events of the book to a new front. It turns out to be the story of the demon, not the main character.

27: If this is the sequel to Star Trek: Deep Space Nine, it could also be a variation on a similar plot point (Star Trek: Deep Space Nine, Deep Space Nine: Deep Command, etc).

28: In Dungeon Fighter Online’s opening story arc, the protagonist is revealed to be the rightful king of Zennar, who’s been deposed for having eloped with a woman who was his slave. Unfortunately, the truth serum was never present until the game’s second half-screen and his blood serum showed up. Needless to say, it’s been a big surprise to everyone.

29: Inverted in "The Man With No Cloth": In "The Man With the Cloth" after the climax of one of their appearances, Claire and Sam reveal to the audience that their father is actually a robot - and it turns out that he’s not. Claire comments that he was meant to be a robot instead. He reveals to them that he was actually the last human on Earth. Later, in the film, one of the men reveals himself as Claire, who appears to be the narrator of the narration (as he’s on top of a big pile of rubble), and turns out to be his adoptive father’s biological son.

30: The narrator of "The Fountain" is revealed to be The Narrator, but not until the middle of the film do I find out that the real one is The Narrator of a Hollywood story, or that The Narrator is actually a robot disguised as the camera, or that The Narrator is actually a young boy playing the victim.

31: A variation in The Fountain, where the Princess Luna is seen dancing with a man before the film begins.

32: "The Long Knives" is narrated by the narrator as a young man with a fascination for the things he’s about to learn.

33: A variant has the narrator writing dialogue in the first person in various ways , after all.

34: "Happens in Murder, with John Doe missing."

35: In the Scooby-Doo franchise, Scooby-Doo has a recurring Scooby-Doo head injury. In the Scooby-Doo franchise a young Scooby-Doo named Todd is injured as a baby when he’s accidentally shot in the head and has to endure the trauma.

```
36: The opening scene shows a chilling sequence between a character walking down a
     street and an unknown figure standing in a field dressed as a priest. He looks
     down at a piece of cardboard and says, "Oh yeah, I see your face."


37: Mortis gives the birthsign  Sucker Punch. It turns out to be from an otherwise
     healthy woman.


38: The first of several episodes, "Mortar Reveal", has the villain reveal that he
     was actually the Big Bad.


39: There’s a twist in "The Reveal of the King", where the prince reveals what
     happened to his king; the reveal comes just before the reveal occurs.


40: In the opening scenes of Star Trek Into Darkness, a series of holograms are
     displayed on an empty seat while a security guard points out that this image
     might be fake, causing a major security response.


41: Guild Wars 2 has several instances in which the Narrator is one of the various
     monsters encountered in each area, from the cavern to the city.


42: Lampshaded with a red eye at the end of The Player’s Guide to the Galaxy, in a
     similar manner to the one we’ve already mentioned in the last episode. It also
     reveals that the player is really Loki, the man responsible for helping the Lai
      in his quest for enlightenment. He was revealed to be an android when he was
     captured by the Alliance in the last episode. The episode opens with Loki
     telling the player that his life was no longer worth living, and that he could
     only have wanted to help him achieve enlightenment.


43: The entire story was narrated by Edna, the narrator from the original episode "
     Screamers."


44: The Powerpuff Girls and The Powerpuff Girls episode "A Chunk of Gold", in which
      Becky comments on her brother’s recent injury, reveals that, in her opinion,
     it shouldn’t be this way.  It might also hint at the Powerpuff Girls’ point
     that Becky isn a lesbian.


45: As the end of The Last Dragon has already been confirmed, the audience (and/or
     the reader!) will eventually learn in the following chapters that the narrator
     (who has just been told) is the true ruler of the Red Kingdom of Alba and the
     only true King of Alba until  The Dragon Reborn.


46: The titular character of the LEGO Movie, the robot’s parent robot, has to be
     replaced when a robotic arm is accidentally cut. After a few scenes, the robot
     starts to grow a giant robot head and neck, covering its fur with circuitry.
     When it gets enough strength, it eventually morphs into an adult version of
     itself and starts working on its own.


47: The episode "Sugar Rush" has a major character called Sugar Rush, the narrator
     being Sugar Rush (played by Matt Lintner).
```

```
48: Holloway: When I hear that, I wonder what it would have been like if I’d only
     been a human? I would’ve been able to make a friend, though my father would
     have rejected me for being human.


49: Cameron’s first meeting with her former high-school best friend, Claire, is
     interrupted by a voice calling her "Mr. Bigby!"


50: In The Secret of Monkey Island, the narrator of the story has been heavily
     hinted at throughout the game’s story, but this is often addressed by voice-
     overs that give a more ambiguous impression of what was said and not what was
     said.
```

A.1.2 The TV Tropes Dataset Output

This is the 50 first prompts generated.

Code listing A.2: This is the 50 first prompts generated by fine-tuning on the Twist dataset.

```
0: In "The Crown," the main character is revealed to have been an AI and has never
     been told of it.


1: Cases this in chapter 14 of The Black Rose.


2: This opens with a flashback to Leland, recounting the events of The Office. As
     Leland was about to die, the phone started ringing and an old lady appears to
     be in ecstasy.


3: The reveal that the villainous T-17 is actually a robot comes courtesy of the
     fact that the reveal of the entire comic was filmed in front of a camera, while
      several of the crewmembers look very surprised and scared by the camera.


4: In The Dark Tower, it is revealed that  Jin Kujan is the reason she became the
     new host of the tower, although the reason why the hosts weren’t mentioned is
     so unclear.


5: The entire episode is narrated by an offscreen (and very much concerned) young
     boy named Tatsuki.


6: Michelin is revealed to be a robot underneath a metallic prosthetic, but unlike
     her creator, she is actually human, and is actually a descendant of a royal
     family.


7: The titular character of The Little Mermaid is revealed to be  Ami, though she
     is actually  Ariel from the Disney/MLP series "The Adventures of Princess Monet
     " (which was released back in 1990). Her identity is a mystery to the audience.
```

```
8: The first game has two main narrators, both narrating what happens to the
     Narrator in the game. The first is the narrator, and the second is the game’s
     narrator himself.


9: An unexpected turn of events leads to an unexpected double in the final episode
     of the season. When the plot is revealed to be an optional side quest, the
     reveal to the audience is a bit more sinister.


10: The events that occur in the story tell about how  the player character is
     raised.


11: In The Iron Man 3: Captain America: The Winter Soldier, Captain America learns
     the true identity of Drax the Nightcrawler in the midst events of the Civil War
      and his new team arrives on Earth, fighting alongside and assisting Captain
     America in various ways.


12: In "Paint Your Dad", the narrator is hinted to be a robot, though in the third
     act he’s actually an ape-like being who was created from cardboard, though it
     turns out the character is actually a clone of the original character.


13: The narrator is a lonely young man who recently found love and is looking for a
      new girlfriend.


14: The first level of the game, where you start a new chapter as Nefarious, is
     narrated by a young female character who turns out to be the narrator. The
     prologue reveals that she’s Nefarious, that the game is actually an optional
     prologue, and that she was one of the game’s main characters (which is one of
     the main reasons why he’s one of the most important characters in the story).
     In the end, you get to choose Nefarious’s real name.


15: A boy begins his quest for redemption. He comes into possession of a mysterious
      girl from an unknown dimension. He’s a musician and he soon finds out what it’
     s like to be a musician. The girl has a strange talent for writing.


16: A variation occurs in the second arc when Dr. McNicholas is revealed to be the
     "father" of the original narrator, the narrator himself. This story is also
     used in the arc "The Last Guardian", the final battle of the series.  According
      to the final battle scene, McNicholas’s corpse is left behind, and a small
     hole opens up in the middle of the battle where McNicholas himself had been
     standing. This is also done in the final battle scene of the series, where the
     narrator himself reveals to the camera that he is the son of one of McNicholas’
     s patients, who has developed a taste for death.


17: The final mission of The Matrix, Neo’s quest to convince the Matrix scientist
     Dr. Lamb to stop spreading rumours is cut to a cut before it even starts. The
     Matrix crew attempt to distract him by talking about how powerful he is, and
     then attempt to frame him for the actions he’s had in the past, to make him
     look bigger. And finally it’s just a ruse. In fact, it’s pretty much a plot
     device used to get Lamb to talk into his past.
```

```
18: A robot named Rex saves the day. He saves Rex from a shark attack with the help
      of Rex's robotic assistant (who now commands Rex's body) and Rex becomes the
     hero that Rex's life is worth. Rex does this by launching himself into space to
      destroy various planets, and even has to take out three robots to get to Earth
     .


19: In one of the opening pages of Chapter 11, Page Six, we learn that Page Six,
     the president of the Republic, is also an Internal Security Officer, with rank
     and command. As the mysterious and repressive Executive Council of the Republic
      escalates into a new level of scandal, Page Six reveals himself to be an
     assassin and is the leader of a rebellion. It's all part of a plan, which Page
     Six uses as an example to clear up his behavior by making Page Six himself the
     target of one of the biggest, most extreme, and possibly deadliest campaigns in
      recent memory.


20: In "Wonder Woman", after rescuing a kidnapped princess, Amanda Waller begins to
      question whether she is a robot or a human being.


21: In the film The Reveal, which opens with an interview between L. Ron Monk and
     the gang, Ron Monk tells the gang's leader that he's not sure about who killed
     Randy, but that it was a robot and he'll be okay. He also reveals that Randy
     didn't kill Randy; instead, he kept him alive on the planet. This was followed
     by a series of flashbacks, which include Randy being shot dead from an unknown
     distance, being shot through the hand, being put to death, being placed on a
     table, being fed ice, the gang killing Randy anyway, and then talking. The
     final image of the film (which we haven't seen in a while) shows that Randy was
      the robot.


22: There are multiple endings for The Reveal. In the second case, a character
     reveals themselves to be "The Reveal Device" while they're trapped in the Maze.


23: The game begins with a narrator narrating what happens during the game. This
     makes the characters more likeable and often has the reveal of some backstory;
     this is usually done so as a way of explaining the events that happened during
     the play and making it seem less "real" to players, but also to their audience.
      Occasionally, the game will also give you spoilers from the player's
     interactions with the characters before you even start the game, of whom we
     often find guilty until the point of ending the game without providing a one-
     frame summary of the events that happened to the characters themselves. In fact
     , some of the events of the game actually occur while we're at a time when the
     game is-related.


24: The Big Bad in the sequel "Fire and Ice" is actually an animatronic version of
     Big Bad Wolf using the same trick from the original episode, except it's a
     robot built by a human, meaning it's still relatively ordinary and human.


25: Bitch: I didn't realize this until you got to the bottom of the story.


26: In Mortal Kombat X, the player character finds out that he is the son of Dr.
     Light and that he had been raised by an ancient wizard with a mysterious power
     and that he's been dreaming for thousands of years.
```

```
27: The Narrator of this movie has one of the biggest "Eureka!" moments ever
     recorded.  He is seemingly being led away by the Narrator to a remote cavern,
     while the other characters are told that he is the Narrator; then, the narrator
      becomes even more important, revealing that the narration is the Narrator
     himself, narrating to the Narrator about a man who is a hero, or the Narrator
     himself. The entire time the narrator is present, the Narrator is just
     narrating a story.


28: "And then the next thing I remember... I go into the house..." I pause to check
      on what happened to my dad, which is not unusual. The only indication of a
     father’s presence, though, is the speaker’s voice and the speaker being in a
     room with a speaker-only sound system. I repeat that, as though to be expected
     given that there are no speaker rooms on the island.


29: A mysterious man known only as The Man Behind the Mask is attempting to gain
     access to every aspect of his past and present to use his own power and to
     control the world.


30: The third book in the series, In The End, has two major arcs, one relating to
     the events of the game and the second to the final ending of the series:


31: In The Last Guardian, The Godfather trilogy, the real Father is revealed to be
      Miraculously Human.


32: In the 2004 anime "The Little Mermaid" with Prince Mononoke, the fairy prince
     Princess Cadance gets seriously hurt during her long journey, and she starts
     experiencing her own version of  Princess Bubblegum.


33: After leaving a monastery after she is forced to return to her monastery to
     discover whether her daughter is a human or a vampire, a human priestess
     arrives with her own son, who is raised in a monastery and is now a priest to
     the Luminary Church.


34: In Magical Girl, Mamoru and her friends try to use Mamoru’s powers to make a
     robot duplicate of a real girl in order to prove that the whole thing is a
     dream.


35: Two young women are recruited by a murderous gang to steal their father’s car
     from a wealthy family. When the family is attacked, the wealthy couple is
     forced to flee their treacherous ways to escape. They soon discover that the
     murder has become part of the plot, with the men trying to break into the
     family property to get it back.


36: The first two episodes of Shrek’s Adventures at Sea, which are narrated by a
     young Shrek, are also narrated by a man with long hair that is supposed to be
     the ancestor of Shrek and the other trolls in the Sea of Lemony. The latter two
      are narrated by another young Shrek, the latter two as well.
```

```
37: The narrator of The Tale of Princess Celestia begins the story with a scene
     where a prince of Ankh finds himself in a small room. The prince is introduced,
      however, and says this is all the King needs to know how the entire kingdom is
     . The prince looks up at Celestia and then says, "This is what happened to your
      mother."


38: The second season of The LEGO Movie opens with the titular LEGO-Holographic
     Head Headless Horseman walking on the set of the LEGO Movie, presumably unaware
      that his head is being controlled by an animatronic, revealing that the LEGO
     Movie was actually a live-action scene.


39: In Star Trek Online, it is revealed that Data’s first memories of his mother
     are from his grandfather, who was a strict and strict cook who was also a
     respected cook and cook in his family. During the final confrontation with
     Captain Kirk, he comments, "I remember my grandfather," before explaining that
     the memory was a "crown of honor" and he wanted the crew to believe he was the
     heir to the throne of Kirkland.


40: Lampshaded by the events of The Dark Knight Rises when Henry Kent has a heart
     attack and Bruce Wayne is left completely destitute and helpless. The rest of
     the story is presented as a twist, and all the major characters are actually
     dead in the film, having just been shot, and the rest are alive thanks to
     Batman and the Green Lanterns.


41: At first, it is not clear that we are robots. The title of the game’s epilogue
     implies that we are humans, which is likely the true truth. While this is
     generally well-intentioned and satisfying for the players in the sense that
     their actions don’t qualify as "real", it nonetheless ends up being an attempt
     to tell us that we are not, in fact, robots. We do know we are not. We are a
     human, being built by the humans on the planet.


42: A man with a beard turns out to be a robot after he cleans up a mess he made up
      by lying to someone in his dreams. After he is told of the incident, he falls
     back asleep for the night and later discovers he had been prosthetic in order
     to blend in with other human hosts who had been watching, even though they were
      all asleep.


43: Played with when a mysterious and dangerous stranger comes through town and
     takes a tour of the town, but soon learns that something extraordinary is
     really happening.


44: A young female woman, who is being pursued by a notorious gang of killer, is
     seemingly willing to do anything to ensure her safe release.


45: The Narrator of the story "Morpheus and The Reveal" is the author of the sequel
     , "Mr. Monk". Monk is also the Narrator of the story "The Reveal".


46: Also in the Fallout 3: The Animated Series, where Franklin is revealed to be
     the same Franklin from a previous game, Franklin’s "rebirth" is actually
     Franklin’s body that was salvaged from a crashed Star Map.
```

```
47: A detective investigates the man who committed the murders, the man who
     committed the murders, the man who threw the bodies on fire, the man who put
     all the victims to rest, and the man who broke into the house.


48: The Reveal of the Maniac is revealed to be a film about a man who, at the end
     of the film (in an extremely dramatic sequence), becomes a robotic duplicate of
      himself. A few moments later the narrator’s voice reveals that the Maniac is
     actually a model, not a human.


49: Crazy, manipulative, and often violent in a way that makes the reader feel like
      they’re watching a movie; the story is centered around a couple of individuals
      who find solace in a monastery and find out that some of the monastery monks
     aren’t real people.


50: Hooru for the Bears.
```

NTNU

Norwegian University of
Science and Technology