MSc Computer Science
Final Project

Generating Video Game Quests From Stories

Manish Kumar Mishra

Supervisor: Dr. Lorenzo Gatti,
Dr. Faiza A. Bukhsh

December, 2023

Department of Computer Science
Faculty of Electrical Engineering,
Mathematics and Computer Science,
University of Twente

UNIVERSITY OF TWENTE.

Contents

Abstract

With the increasing interest in story based video games, there is a surge of new titles based on existing stories or narrative worlds. Game development is a notoriously challenging job and this project aims to streamline the workflow by automating quest generation from existing narratives. The narrative plot points and the relations between the included characters, locations and objects form the crucial story elements. This thesis delves into the incorporation of world knowledge as knowledge graphs (KGs) in quest generation in video games from story plot points. Knowledge graphs capture the intricate dependencies between the entities in a narrative world and this information is leveraged in generating quests. The primary focus lies in examining the impact of incorporating KG data on the quality and relevance of generated quests, an area that remains relatively unexplored in the field of automated game design. Through a methodical approach, this study evaluates the impact of incorporating knowledge graphs for quest generation. The findings reveal a significant improvement in quest quality with the inclusion of KG data when compared to those generated without the KG data. The similarity of generated quests with the actual quest is also assessed by changing the amount of included KG data.

Keywords: LLMs, Knowledge Graphs, Video games, Quests, NLG, Stories, Llama2

Chapter 1

Introduction

Traditionally, storytelling has been done using static or non-interactive media such as books, movies, music and TV shows. Recently, many story-based computer games have become a cultural phenomenon, with some even being inspired by existing narratives. The COVID-19 pandemic witnessed a surge in the popularity of video games as people sought solace and entertainment during lockdowns and stay-at-home orders [1]. This unprecedented demand encompassed not just online games but also single-player experiences characterised by immersive gameplay and captivating narratives.

Video games are considered a modern form of storytelling with an engaging and interactive experience. Recent video game titles such as The Witcher series, Cyberpunk 2077, Mass Effect, BioShock, and Assassin's Creed have been directly influenced by established narratives in books or historic events [2, 3, 4, 5]. Some best-selling fictional novels, such as The Lord of the Rings and the Harry Potter series, have inspired multiple video game adaptations. The immense success of The Last of Us as a video game has spurred the development of a television series of the same name, providing an alternate medium to narrate the game's story.

Telling a long and compelling story is generally done by breaking it down into smaller parts. Books achieve this with chapters, whereas TV shows break the story down into episodes and seasons. The most common way for video games to create an interactive storytelling experience is by breaking down the main narrative into smaller, objective-based "**Quests**". Each quest plays out a single or a chain of connected events in the game that progresses the overarching narrative. Quests allow the players to explore the game world and engage with the storyline in a much more personalised manner. Role Playing Games (**RPGs**) and their online counterparts - Massively Multiplayer Online RPGs (**MMORPGs**) are widely considered genres for video game story-telling using quests, wherein players assume the role of the narrative's main character. The game world is often populated with non-playable characters (**NPCs**) who have personalised backstories and provide details for story quests or side quests.

Although challenging, creating a fun and interactive storytelling experience in games has proven to be a successful and enjoyable experience for players worldwide. Crucial story elements, such as plot points, characters, locations and objects, must be identified to create an interactive experience which stays true to the narrative [6, 7]. In the context of this research, plot points in a story are events that occur in the narrative that progress the narrative forward and possibly change the state of the game world. The narrative entities, i.e., the characters, locations and objects, are either directly or indirectly connected to these plot points or events. The relations between these entities narrate how the story or the quest plays out.

This research project aims to streamline the workflow of generating quests during game development by automating quest generation for new video games based on existing narratives. For the scope of this project, existing narratives refer to stories from books, movies, TV shows, and lore-heavy video games that do not inherently feature quests. The first step towards automating quest generation from narratives is to enable generative language models to understand the narrative world. The plot points serve as the essential representation of the narrative flow of the story [7]. Meanwhile, the story entities and their relations provide further contextual information with respect to the plot points.

Current state-of-the-art Large Language Models (**LLMs**) such as *GPT* and *BERT* showcase exceptional natural language understanding abilities for language generation tasks. However, LLMs often tend to fantasise when it comes to knowledge-based or contextual generation tasks, frequently generating illogical or incorrect text. Researchers use the term "hallucination" to refer to this phenomenon. Context-based representations in the form of knowledge graphs have proven to improve Natural Language Generation (**NLG**) tasks by enabling LLMs to retrieve relevant information for context-aware text generation [8, 9, 10, 11]. Knowledge graphs (**KGs**) have also been used to model narrative worlds in creative text generation tasks, as they capture and visualise the intricate relationships between various entities of the world [7, 11, 12].

This project aims to understand the impact of integrating data from world knowledge graphs in quest generation. Essential plot points from the story act as the primary input to the quest generation model, providing it with the narrative flow. The knowledge graph data will be integrated into the input of the quest generation model, and the impact of varying levels of this integration will be assessed. The quest generation model is a fine-tuned version of the pre-trained Llama2 model by Meta [13]. With the discussions laid out, the scope of this thesis is to understand the impact of incorporating game world knowledge from knowledge graphs in quest generation using story plot points. The following research questions have been formulated to guide this research project:

> **RQ1**: *How does the integration of world knowledge affect the quality of quest generation using LLMs from narrative plot points?*

World knowledge here refers to the entities present in the narrative world along with the relations between them. The plot points capture the narrative flow of the story. To adequately address the research question, a literature review is conducted for gathering requirements, forming the basis for the design of a robust quest generation framework. Following this, experiments to evaluate the quality of generated quests and the effect of adding world knowledge through a knowledge graph are built. To further guide this research, the following sub-questions will be explored:

> **RQ1.a**: *How similar are the quests generated with varying amounts of contextual information provided by world knowledge graphs?*

The world knowledge graph would contain information about every entity and the relations between them. Adding information irrelevant to the plot points concerning the quest might only sometimes be helpful. In addressing this sub-question, varying amounts of information from the world knowledge graphs are incorporated with the narrative plot points for quest generation, and the resulting observations are documented.

> **RQ1.b**: *What effect does incorporating knowledge from knowledge graphs into LLMs have on improving the quality of generated quests in terms of story relatedness, coherence, contradiction and progression value?*

The quality of quest generation using LLMs is measured using a survey in which human participants are asked to judge the generated quests on the mentioned criteria. Relatedness and coherence measure how well the generated quest represents the story, progression value measures how well the quests follow the story, and contradiction measures the level of hallucination by the quest generation model or how incorrect the quests are with respect to the story.

In the remainder of this thesis, an overview of fundamental concepts about quests in video games, natural language generation, knowledge graphs and large language models is provided in chapter 2. Chapter 3 reviews existing literature and previous work in the field, highlighting their relevance and contributions to this project. Chapter 4 outlines the proposed approach, including the implementation details for the methodology used to answer the research questions. Chapter 5 discusses the results and the evaluation process answering the research questions. Finally, chapter 6 concludes this research work and presents some possible future extensions to this work.

Chapter 2

Background

This chapter provides an overview of quests in video games, Knowledge Graphs (KGs), Natural Language Generation (NLG) and Large Language Models (LLMs). It lays the fundamental knowledge required to understand how these concepts are used in unison to generate quests in this project. Firstly, the definition and structure of video game quests in academic research are presented. Next, the concept of Knowledge Graphs is explored, which serves as the knowledge base for the proposed quest generation framework. Further on, the recent advancements in NLG and LLMs are presented, which serve as the foundation for the quest generation model.

2.1 Quests

Quests are often regarded as missions that the player undertakes to eventually reach the goal of the main storyline, often accompanied by in-game rewards. In a technical sense, quests can be considered as soft rules that a game enforces, comprising a set of tasks that the player must complete in order to achieve the game's final objective [14, 15]. Quests are designed to allow the players to engage in smaller stories, which together form the overarching narrative. In this section, the structure and design patterns of quests, as seen in academia, are identified - which can be used to employ modern generative models for quest generation and control the outputs desirably. Figure 2.1 shows a screen grab for a quest from the game "The Elder Scrolls V: Skyrim"[1], where the player meets the quest giver, followed by the in-game quest log which displays the title, description and the objectives of the quest.

In their research, Doran and Parberry [16] analysed over 750 manually crafted quests from four different MMORPGs. They discovered that quests typically adhere to a standard structure, with variations primarily in the game setting and context. Each quest is composed of either an NPC dialogue or a description that establishes three key elements: (a) the *motivation* for the quest - the most critical concern that the quest giver seeks to resolve, (b) a *sequence of tasks* that the player must accomplish, and (c) the *reward* upon quest completion, which compels the player to complete the quest.

Yu et al. [17] conducted interviews with fifteen professionals from the gaming industry who have substantial experience in quest development. The objective of

[1] https://elderscrolls.bethesda.net/en/skyrim10

Figure 2.1: A screenshot of the first quest from the game "*The Elder Scrolls V: Skyrim*". The first image shows the initial interaction between the player and the NPCs, who explain the current situation to the player after they are finally awake. The second image shows the in-game journal entry for the quest, with the title "*Unbound*", followed by the description and the quest objectives

these interviews was to establish a formal definition of a "quest" within the context of video games. According to their findings, a quest denoted as Q can be defined as a collection of partially ordered tasks, $t \in T$, and a distribution of rewards on completion of the tasks, represented as $d(R)$. A task t is considered complete when it is executed by the player and certain conditions, represented as C, are satisfied. Furthermore, they define a monitoring system, denoted as M, that is responsible for checking the state of the game to confirm whether tasks within the quest have been completed.

The World of Warcraft quests database published by van Stegeren and Myśliwiec [18] exhibits a similar structure for the quests. The quests are extracted and compiled from the WoWheads quests database[2], an online forum and database for the game "World of Warcraft". The quests contain the following fields of interest. The *title* of the quest; the *objectives* - a set of tasks that the player has to complete; a *description* of the quest, which includes the backstory and the motivation of the quest giver; and finally, the *completion* message that the player gets on completing the quest.

The quest definitions seen above offer a comprehensive understanding of the structure and components of quests in video games. These structures include the **title**, the **objective** and a **sequence of tasks**, providing a basic framework for quest design. In this project, the generated quests follow a similar structure, consisting of *title* of the quest, the main *objective* that the player has to achieve and a set of *tasks* to perform to reach the objective mentioned above. Rewards are not included, as the focus of this project is on the creation of quests from a narrative and story-telling perspective.

2.2 Natural Language Generation

Quest generation is a sub-task of generating natural language output, following the format described earlier in section 2.1. Therefore, this research would leverage recent state-of-the-art Natural Language Generation (**NLG**) techniques. NLG is a subtask within natural language processing (**NLP**) that involves the generation of text from textual or non-textual input. Automatically generating natural language is a challenging task where the domains of computational creativity, linguistics and artificial intelligence intersect [7, 19]. Specific tasks such as generating stories or quests require the generated text to meet desirable constraints, including output structure, that needs to be controlled [20].

Initial approaches to NLG were based on simple NLP techniques and rule-based models, which followed templates and generation pipelines to produce output. For example, the early chatbot application **Eliza** [21] used pattern matching to answer questions in the style of a psychotherapist. The prototype quest generator by Doran and Parberry [16] also uses templates to fill in action phrases in the generated quests. However, these models did not possess the ability to learn from new data and had a limited understanding of language.

Recent methods use probabilistic modelling of word embedding to predict the next word in the sequence of text. These models can be trained on available data to enhance their understanding of language. Models using Long Short-Term Memory (LSTM) [22] cells and Gated Recurrent Unit (GRU) [23] have shown promising results. These models are based on recurrent neural networks (RNN) designed to store inputs from earlier time steps (previous words in the sentence), enabling the retention of contextual details across longer sequences. However, these models still suffer from vanishing context in longer sequences such as story plots. Subsequently, the use of Variational Auto-Encoders (VAEs) gained traction as deep generative models [20]. VAEs consist of an encoder model, which learns the distribution space of the real data, and a decoder model, which generates new data by sampling a random point within the learned real data space [24].

Building upon VAEs, the attention-based **Transformer** models [25] have emerged

[2] https://www.wowhead.com/quests

as a breakthrough, surpassing all previous state-of-the-art models. Initially developed to enhance machine translation [26], the attention mechanism utilises alignment scores to establish the correlation between generated output sequences and their corresponding input sequences. These scores let the model know which token in the input sequence needs "*attention*" for the next output token to be generated - making them suitable for a variety of downstream NLG tasks. The transformer model uses positional data encoding instead of fixed-length context windows to solve the vanishing gradient problem over long input sequences. Additionally, recent transformer implementations improve upon VAEs by adding a *self-attention* mechanism, enabling the model to weigh the significance of attending to individual tokens within the input sequence to influence the output [25].

2.3 Large Language Models

With the rapid advancements in hardware, transformer-based language models have evolved into Large Language Models (LLMs), characterised by their extensive model parameters and colossal training data. The academic and industrial research communities have contributed numerous LLMs that have already been trained on a myriad of tasks such as text completion, common-sense reasoning, reading comprehension, mathematics, and coding. These pre-trained LLMs are able to capture the intricate details of language and have performed very well in multiple Natural Language Understanding (NLU) tasks. Some notable examples include OpenAI's GPT-2, GPT-3 [27, 28], Google's BERT [29] and T5 [30], Meta's BART [31], LLaMa [32] and Llama2 [13].

Many studies have shown that amplifying the scale of language models has not only led to an improvement in understanding language, but after a certain threshold, sufficiently large models potentially exhibit unique language capabilities absent from language models of smaller sizes [33]. However, some researchers argue that such improvements might actually be a result of the choice of metrics in the research rather than fundamental changes in model behaviour due to scaling of their size, as demonstrated by [34]. Nonetheless, current-day state-of-the-art LLMs exhibit remarkable proficiency in generating grammatically correct natural language based on the input or a prompt by the user.

However, the input text or the user prompt does not always provide all the required knowledge to generate the desired output, causing the LLMs to occasionally suffer from hallucinations, generating text which is factually incorrect or out of context [33, 35]. The ability to *fine-tune* these models for specific use cases helps in mitigating such issues. Providing them with more contextual information about a given topic can enhance their performance and generate more accurate and relevant results [36].

2.3.1 Fine-tuning LLMs

Fine-tuning is a transfer learning technique commonly used to adjust Large Language Models (LLMs) for specific downstream tasks, often with a smaller task-specific dataset. It involves updating the parameters of an LLM by training the model on data for the task at hand. Fine-tuning significantly reduces the training time and data requirements as compared to training a model from scratch for a specific task. However, fine-tuning LLMs demands high memory and computational power requirements, and with increased model parameter sizes, professional-grade equipment is required, even for smaller tasks. Another drawback observed while fine-tuning LLMs is *catastrophic forgetting* [37], in which the model drastically forgets previous information while learning new information.

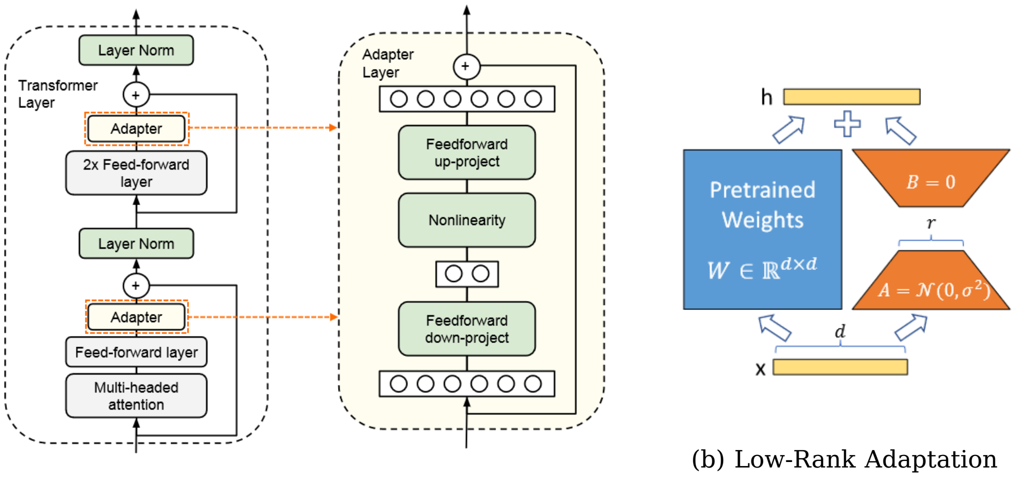
Numerous strategies have been proposed for efficient fine-tuning to overcome the issues faced during fine-tuning LLMs. Methods such as *Prompt Tuning* [38] and In-Context Learning (ICL) [27, 28] augment the inputs given to the model with contextual information either implicitly as tunable embeddings (soft prompts) - where the inputs are slightly modified to include information in a human unreadable format; or explicitly as a prefix to the input text (hard prompts). These inputs are then optimised for the task at hand during fine-tuning while the model stays frozen, meaning that the model parameters are not updated. These prompts are optimised as a part of the fine-tuning process to condition the LLM to perform the task.

Another recently proposed approach for task-specific adaptation of LLMs is Parameter Efficient Fine-Tuning (PEFT) [39]. This approach involves the modification of only a minimal subset of parameters within the LLM. PEFT techniques have demonstrated comparable performance to comprehensive model fine-tuning while only tuning a minuscule amount of the total parameters of the model. This approach enables efficient task-specific adaptation while preserving the majority of the pre-trained model's parameters, mitigating catastrophic forgetting and also requiring lesser computational resources.

Injecting and training task-specific adaptors work as a computationally efficient alternative approach to fine-tuning. Adaptors are smaller networks within the transformer module which are trained during fine-tuning while keeping the parameters of the LLM frozen. The proposed injection of adaptor network [40] is shown in fig. 2.2a.

Low-Rank Adaptation (LoRA) [41] builds on this, providing a more efficient and effective alternative to traditional fine-tuning processes. LoRA utilises rank decomposition of matrices, which simplify complex operations on high-rank matrices by splitting them into constituent matrices and applying the operations on the smaller matrices. In the context of LLM fine-tuning, LoRA injects trainable adaptors with lower rank weight matrices in the otherwise frozen LLM layers. These matrices are decomposed from the pre-trained weights and allow for a much more compute-efficient and low-memory fine-tuning while performing the same or better than traditional fine-tuning methods. fig. 2.2b shows the proposed parameter decomposition methodology from [41].

There have been multiple efforts at compressing LLMs to reduce the memory footprint during model training and inference by reducing the bit precision required to represent model parameters [42, 43, 44, 45], allowing usage of models on low memory devices. This method is called *quantisation*, and together with the PEFT methods, substantially reduces the computational power and memory requirements for training and inferring from the models. **QLoRA** [46] is one such method, which enables fine-tuning and inference from LLMs on consumer-grade hardware by loading the LLM using quantisation techniques and further applying Low-Rank Adaptation (LoRA).



(a) An adaptor network injected into a transformer module [40]

(b) Low-Rank Adaptation (LoRA) of pre-trained network parameters [41]

Figure 2.2: Parameter-Efficient Fine-Tuning (PEFT) methods

2.4 Knowledge Graphs

A knowledge graph (KG) is a graph-structured data model to store and convey knowledge within a context. KGs are composed of interconnected nodes and edges, where nodes represent entities of interest within the knowledge base, and edges denote the relationships between the entities [11, 47, 48]. By organising contextual data in a structured format, KGs offer significant potential to enhance downstream NLP tasks, particularly in the domain of natural language generation [11, 36]. In the context of this project, knowledge graphs will be constructed to represent the world depicted in the narrative, thereby serving as a contextual knowledge base for the quest generator LLM. The KG will contain objects, locations and characters of interest as the entities and the edges will define the relations between said entities.

Chapter 3

Related Work

The first step to answer the main research question at hand:

> **RQ1**: *How does the integration of world knowledge affect the quality of quest generation using LLMs from narrative plot points?*

- is to establish a theoretical foundation based on background research of existing literature. This chapter reviews previous research efforts in story and quest generation in section 3.1, along with a focus on utilising KGs and integrating KGs into Large Language Models (LLMs) in section 3.2. Related work in constructing Knowledge Graphs (KGs) from text corpora and their applications in text generation tasks are also presented in section 3.3. Additionally, methods used for evaluating the generated text and examining approaches to assess the quality are discussed in section 3.4. Finally, the conclusions drawn from up-to-date research are outlined, along with the fundamental aspects of the quest generation framework designed in this project in section 3.5.

3.1 Story & Quest Generation

Doran and Parberry conducted one of the earliest academic research works on quest generation [16]. As mentioned in section 2.1, the authors analyse the structure of over 750 human-written quests from four MMORPGs. Further, they classify quests into nine categories based on the underlying motivation. Building on their analysis, they create rules that govern the sequence of actions based on these motivations along with action pre- and post-conditions. Using these rules, they create a prototype quest generator that uses generative grammar to produce quests using the established rules.

Choi et al. [49] propose a two-network model to generate stories with correct grammar and context flow. The first network, RNNED, uses an RNN-based encoder-decoder network which learns sentence vector representations. The second network, RNNSG (RNN for story generation), learns the flow of contents in stories as sentence vectors and generates novel stories as sentence vectors. The generated vectors are then decoded into text using the decoder of the RNNED model.

Affect-LM is another RNN-based language model proposed by Ghosh et al. [50] that results from an extension to an LSTM model, with two additional parameters to train the language model: affect category and affect strength. However, modern transformer-based LLMs for text generation have become the paradigm for NLG tasks, outperforming RNN-based models [20].

Van Stegeren and Myśliwiec [18] worked on tuning GPT-2 to generate quest-giver dialogues for World of Warcraft. They train the model by creating a structured input that contains tags for the title, objectives, description, and end of the text. The title and objectives serve as the input, and the description as the output. The model is given a new title and objective and prompted with the title tag to generate a new description for the given inputs. The GPT-2 model has a parameter called "temperature" to control the randomness of the outputs. The authors adjust the temperature to influence novelty in the generated outputs while maintaining language coherence. To evaluate the dialogues produced by their model, the authors use a human survey and find that the model outputs perform very close to hand-written alternatives in terms of content, novelty and coherence. However, their model was fine-tuned on the specific World of Warcraft quests dataset and would benefit from further generalisation for all kinds of RPGs. This project, on the other hand, aims to generate quests from story and plot points.

Quest-GPT-2 is a fine-tuned variant of GPT-2 introduced by Vartinen et al., trained to generate quest descriptions from quest inputs [51]. They fine-tune a GPT2 pre-trained model for the purpose. Their evaluation results show that while the generated quest descriptions show promise, they sometimes exhibit questionable logic, repetition, poor grammar, and inclusion of unnecessary information. The authors acknowledge that employing larger and more advanced models could potentially lead to higher-quality results. The authors also published a dataset containing 978 quests extracted from six RPGs. This dataset has proven to be rather valuable for this project due to the information that it contains. More details follow in section 4.1.

Al-Nassar et al. introduce a prototype game called QuestVille, which combines rule-based procedural quest generation with text generated from BERT and GPT-2 [52]. In their approach, the quest generation process involves several steps. Firstly, a random NPC name is selected from a pre-defined list of real-world names. Then, a quest prompt template is selected from a pre-defined list of templates, which contains masked tokens to be filled in with outputs from BERT. The template also contains placeholder tokens that the selected NPC name will replace. The prompt is then finalised by replacing the masked token with the output generated by BERT. Finally, the prompt is provided as input to GPT-2 to generate the final quest. To evaluate the quality of the generated quests, the authors perform manual assessments, considering grammatical correctness, thematic relevance, coherence, and the presence of a clear motivation in the output. They observe that the inclusion of NPC relation prefixing guides the model to produce more favourable results by respecting the provided relations. The authors note that the model occasionally generates high-quality quests, and they find that larger models tend to yield better results.

Ammanabrolu et al. [53] introduce a framework to generate creative quests in text-adventure video games. The authors create a cooking game in which the player needs to find ingredients in the game world based on the recipe provided in a quest. They compare a Markov chain model with a neural-model-based approach. For the Markov chain model, a KG is constructed from the ingredients based on how often two items occur together in different recipes. Using this KG and an ingredient as a starting point, instructions are generated by picking subsequent ingredients using a probabilistic graph walk algorithm. The neural network-based approach consists of two models. The first LSTM model is trained to give a set of ingredients given a starting ingredient. The next is a GPT2 model fine-tuned on generating instructions, given the ingredients as input. They found that the neural network-based approach required less domain knowledge to be trained and produced more coherent quests. In contrast, as evaluated by human subjects, the Markov chain model produces more surprising and novel quests.

These studies show that language models excel at generating quality natural language outputs. It also portrays the need for more research in the field of quest generation using LLMs. Video game content generation by [18, 51] focus on generating NPC dialogues or quest descriptions. Moreover, prominent works done on quest generation including [16, 52, 53] use rule or grammar-based methods to create tasks in quests. In contrast, this project will utilise LLMs to generate quests from story elements.

3.2 Language models integrated with Knowledge graphs

With the increasing capabilities of extending existing language models, there has also been promising work which directly integrates knowledge graphs into language models to improve the quality of generated text.

Logan Iv et al. [36] significantly improve the performance of an LSTM language model by incorporating a local knowledge graph within the model. The local KG grows dynamically, appending facts as they are encountered. While generating an entity token, the Knowledge Graph Language Model (KGLM) takes the parent entity and the generated relation into consideration, decides either to render a new entity missing from the local KG while adding the entity to the local KG or to render a fact from the local KG, or to render a token from the standard vocabulary. This process allows the model to select relevant facts from a knowledge graph based on the context of the application. In addition, the model can render unseen yet correct information while generating out-of-vocabulary words. The authors show that KGLM significantly outperforms larger language models in factual sentence completion and context-aware language modelling, and the outputs can be easily controlled by supplying relevant facts, which the model learns.

Liu et al. propose a novel pre-trained generative language model Knowledge Graph BART (KG-BART) [10]. KG-BART utilises the commonsense knowledge augmented in a ground truth knowledge graph and the learned language representation from the BART language model for commonsense reasoning. In KG-BART, the transformer modules of the language model attend to the knowledge graph along with the textual input. The model uses ground-truth KGs as input and five randomly picked unique entities from the grounded KGs. These entities form a "concept set". The model uses the concept set and the KG to generate a sentence by relating the concepts. Their results show that KG-BART significantly outperformed the state-of-the-art pre-trained models on generative commonsense reasoning by integrating knowledge graphs for context-aware text generation tasks.

The Knowledge-enabled BERT (K-BERT) model introduced by Liu et al. is capable of infusing domain-level knowledge into BERT [9]. This infusion enables the model to solve knowledge-driven problems that often require experts. K-BERT adds a new knowledge layer before the other neural network layers of the BERT model. The knowledge layer generates a sentence tree in which each matched entity between the input sentence and the KG is branched out to all the known facts with the entity in it. The authors show that their model significantly outperforms the base BERT model in twelve open and domain-specific NLP tasks.

Chen et al. introduce the concept of Knowledge-Grounded Pre-Training (KGPT) for data-to-text generation tasks, as outlined in their work [54]. This approach involves pre-training a generative language model while incorporating structured data in a custom dictionary format. In this format, the first entity in a triplet forms the key, and all the remaining (relation, entity) tuples for that entity are the values. To effectively encode these structured sentence trees, they use two distinct methodologies: employing a graph encoder via a graph neural network and a sequential encoder utilising the transformers model [25]. The decoding phase of their model also leverages the transformer architecture, facilitating the generation of responses from the encoded sentence trees. In comparative evaluations, both the graph and sequential KGPT models have demonstrated notable enhancements in performance, especially in tasks related to fact generation and knowledge graph completion.

Grabska-Gradzińska et al. present a novel approach in the field of video games, where they propose a method for generating layered hierarchical graphs from the game world state and represent it in JSON format [48]. The authors' work involves the creation of two types of graphs: the world state graph and the gameplay graph. The world state graph contains the locations, characters and items in the game world at a given state. Meanwhile, the gameplay graph models a player's series of actions during the gameplay that changes the world state. The main focus of their project was to aid storywriters in managing the impact of story elements on the game world and to allow for an on-the-fly procedural generation of world narratives. By collaborating with artists and game designers, the researchers could visualise the unfolding narrative through animated illustrations while also enabling the identification of potential dead-end paths or sequences of actions that prevent reaching the end of the narrative. Their research demonstrates how the narratives plotted as knowledge graphs capture the intricate details of the narrative world, from the characters, locations and items in the world to the influence of actions and plot points on the narrative flow.

Using knowledge graphs to model narratives and leveraging large language models for NLG tasks presents a promising approach. The research outputs mentioned in this section show that LLMs equipped with knowledge graphs as contextual databases can generate high-quality outputs that align with the input text and context. In this project, this translate to generating quests that better align with the background narrative's plot points and entities, ensuring coherence and relevance. However, these studies predominantly involve the pre-training of custom generative models tailored to specific tasks. In contrast, the objective of this project is to utilise pre-existing, pre-trained language models and integrate world knowledge from KGs for the generation of quests.

3.3 Knowledge Graphs

This section explores some traditional rule-based knowledge extraction systems and modern approaches using language models to construct knowledge graphs (KGs) for extracting knowledge from text.

Rule-based methods have been used to extract knowledge from text corpus in older knowledge extraction systems. Stanford's OpenIE is a widely used rule-based tool designed for extracting entity-relation triplets from textual data [55]. OpenIE parses a text corpus and identifies the relationship between entities in a sentence from the text. In its current version, OpenIE 5.1 combines several leading-edge extraction methods [56, 57, 58, 59] and is often used as a first approach tool to extract relations to construct KGs [8, 60, 53].

Recently, constructing knowledge graphs using language models has also been an active area of research, providing promising results. Wang et al. compare the performance of BERT and GPT-2 for constructing knowledge graphs using the attention matrix generated by the transformer-based LLMs for an input sentence. They find that BERT provides better results than GPT-2 at similar model sizes and that larger models provide higher quality outputs [60].

Petroni et al. explore the abilities of BERT without fine-tuning and find that the pre-training helps BERT learn specific facts and, thus, makes it an excellent unsupervised question-answering system [61]. Building on this, Heinzerling and Inui [62] conduct an intensive study on the ability of LLMs to store and query facts and conclude that LLMs can efficiently be used as knowledge bases by appropriately paraphrasing the queries passed to the model.

Ammanabrolu et al. [8] construct the narrative world set by a story as a knowledge graph. They use an OpenIE-based approach and a neural approach for this task. In the neural network approach, the authors use a question-answering AskBERT model by providing the entire narrative as the context. Then, relevant questions (e.g., "*Where is a location in the story?*") are asked to the model to extract entities, and further questions about these entities are asked to extract relations between them. The constructed KG is then completed by generating flavour-text descriptions of entities using a GPT-2 model fine-tuned on plot summaries of stories from the same genre. In their experiments, they find that the world KG generated by the neural model is more coherent and maintains the world theme more consistently.

Although not the main focus of this research, creating a knowledge graph from a narrative is essential. Therefore, as a side-quest to the primary goal of the project, the AskBERT model [8] is used to create world knowledge graphs for quest generation. The model has shown promising results in the construction of KGs from narratives. Further on, the open-source nature of the model and the easy-to-use interface allow for customisation to better align with the requirements of this project. Exact implementation details are outlined in section 4.1.2.

3.4 NLG Evaluation

Evaluating creative output, such as generated stories or quests, is difficult as evaluation metrics struggle to keep pace with advancements in generative models. Traditionally, statistical text similarity scores, such as ROUGE-N, BLEU, METEOR and CIDEr, have been used to assess text similarity [7]. These metrics compare generated output with a target baseline by measuring *n-gram* matches between texts. *n-grams* are contiguous sequences of n tokens in texts. ROUGE has been extensively applied in evaluating machine translation, text simplification, and summarisation models. To evaluate the RNNED model and RNNSG models, [49] uses ROUGE-N scores. Another statistical model used to evaluate creative outputs is *Perplexity*, which measures how well the evaluating model predicts the next token, given the previous output.[36] use this score to measure their KGLM model. However, these n-gram match-based methods do not evaluate the meaning or sentence structure or map well to human judgement[7].

Such models also often fail to match paraphrased texts robustly and capture distant dependencies while penalising semantically critical ordering changes. Improving upon these models, [63] propose the **`BERTScore`** metric, which calculates text similarity using sentence representations provided by a pre-trained BERT model. The authors' experiments show that `BERTScore` correlates better with human judgements and is more robust to challenging examples when compared to n-gram-based metrics.

Given the subjective nature of evaluating creativity, user surveys using Likert scale ratings are also commonly employed for text evaluation [64]. Likert scale is a uni-dimensional scale capturing respondents' agreement or disagreement on a given topic. Van Stegeren and Myśliwiec created an online survey to evaluate the output quest descriptions. The participant users were asked to rate the outputs on a 7-point Likert scale based on *language quality*, *coherence* and *creativity* [18]. Similarly, Vartinen et al. ask participants to rate generated quests on a 4-point Likert scale regarding their desirability for inclusion in a game [51]. Ammanabrolu et al. deployed their created game on an online platform and gathered user feedback on *coherence* and *creativity* (based on the factors *novelty*, *surprise* and *value*) on a 7-point Likert scale [53]. In their subsequent work, Ammanabrolu et al. asked users to rate the game based on *interestingness*, *coherence* and *resemblance to the genre* [8].

In line with these approaches, this project employs a similar user survey, requesting participants to evaluate the output quests based on *relatedness*, *contradiction*, *coherence* and *value* with respect to the story. *Relatedness* and *coherence* assess the accuracy with which the generated quests follow the story, which *progression* value evaluates the extent to which the quests adhere to the storyline development. In contrast, *contradiction* gauges the level of hallucination by the quest generation model, or the degree to which the generated quests deviate from the actual narrative. Along with the user survey, `BERTScore` is used to understand how similar the generated quests are to the actual quests with varying amounts of KG data. The evaluation criteria and the obtained `BERTScore` values are discussed in detail in chapter 5.

3.5 Next Steps

In this chapter, recent advancements in the field of story and quest generation, along with the usage of knowledge graphs was presented. It can be seen that most work in video game content generation are focused on creating quest descriptions or in-game flavor text. Existing quest generation frameworks employ either rule or grammar based methods. Moreover, works that use knowledge graphs as contextual information in generating quests is lacking. With these discussions laid out, this project proposes a quest generation framework which works with the combination of three elements:

E1 A defined set of *plot points* which together form up the game's narrative;

E2 A *knowledge graph (KG)* which encapsulates the crucial entities from the game world, including the characters, locations and objects, along with the relations between these entities; and

E3 A *quest generation model* which will generate quests from the story components - the *plot points* and the *knowledge graph*.

In the following chapters, the steps taken to gather and establish these three elements are outlined in detail, along with the process of utilising them for quest generation. Furthermore, to evaluate the quality of the generated quests, this project employs a user survey, requesting participants to evaluate the output quests based on *relatedness*, *contradiction coherence* and *value* with respect to the story.

Chapter 4

Implementation

This chapter provides a comprehensive overview of the research methodologies and design strategies used to address the main research question at hand:

> **RQ1**: *How does the integration of world knowledge affect the quality of quest generation using LLMs from narrative plot points?*

To this end, along with the previous research from chapter 3, a quest generation framework is designed which relies on three elements:

E1 A defined set of *plot points* which together form up the game's narrative;

E2 A *knowledge graph (KG)* which encapsulates the crucial entities from the game world, including the characters, locations and objects, along with the relations between these entities; and

E3 A *quest generation model* which will generate quests from the story components - the *plot points* and the *knowledge graph*.

Section 4.1 describes the dataset and pre-processing steps to gather the first two elements - the *plot points* & the *knowledge graph*. Section 4.2 outlines the design overview of the third element - the *quest generation model*. Finally, Section 4.3 presents the experimental setup established using the quest generation framework.

4.1 Dataset

The dataset curated and made publicly available by [51] aligns closely with the first two fundamental elements of the proposed quest generation framework: (**E1**) A defined set of *plot points* which together form up the game's narrative; and, (**E2**) A *knowledge graph (KG)* which encapsulates the crucial entities from the game world, including the characters, locations and the objects, along with the relations between these entities.

This dataset is a comprehensive collection of 878 quests, sourced from a selection of five role-playing games (RPGs), coupled with an additional 100 manually crafted quests for Minecraft. The RPGs in question - Baldur's Gate, Baldur's Gate 2, The Elder Scrolls IV: Oblivion, The Elder Scrolls V: Skyrim, and Torchlight 2 - share a common theme rooted in medieval and high fantasy settings. The composition of the dataset, including the number of quests per game and the average word count, is detailed in table 4.1. "The Elder Scrolls: Skyrim" contributes the highest number of quests (389), while "The Elder Scrolls: Oblivion" has the lengthiest average quest description at about 136 words. Overall, the dataset includes 978 quests with an average description length of 103 words, demonstrating a diverse and rich source of narrative content for KG construction and quest generation.

Game	**Quest count**	**Avg description word count**
Baldur's Gate I	100	93
Baldur's Gate II	94	128
The Elder Scrolls: Oblivion	215	<u>136</u>
The Elder Scrolls: Skyrim	<u>389</u>	99
Torchlight II	80	76
Minecraft	100	84
Total	**978**	**103**

Table 4.1: Details of the video game dataset by [51]

4.1.1 Structure of the dataset

This dataset is provided as a list of JSON entries, with each entry containing the information for a quest. The attributes of each entry in the dataset are as follows:

- `name`: Name of the quest
- `quest_giver`: The character who gives the quest to the player
- `objective`: The main objective to achieve in the quest
- `first_tasks`: The set of actions to perform to fulfil the objective of the quest
- `first_task_locations`: The location(s) where the tasks can be completed
- `rewards`: The rewards (object) obtained on completion of the quest
- `characters`: Other important characters related to the quest
- `locations`: Other important locations related to the quest
- `enemies`: Enemies that the player will face during the quest
- `items`: Other objects related to the quest
- `tools` or `plot_points`: The plot points that make up the quest's background story
- `groups`: Important groups or factions related to the quest entities
- `description`: A description of the quest shown to the player, primarily as the `quest_giver` dialogue

The `tool` field contains the plot points from the story that forms the quest [51], which forms the first element for the quest generation framework: (**E1**) A defined set of *plot points* which together form up the game's narrative. As these plots are fundamental to the framework, entries without any `plot_points` are filtered out from the dataset, removing 207 quests from the corpus.

The dataset authors have also identified and listed the related entities, such as the characters (from the attributes `quest_giver`, `characters` and `enemies`), locations (from `first_task_locations` and `locations`) and the objects (from `items`) [51]. These entities partially constitute the second element, the *knowledge graph*. The quest displayed to the player in the respective game is captured in the `title`, `objective` and the `first_tasks` fields in the dataset. This quest data will act as gold data for the quest generation framework. The dataset also contains the quest description in the `description` field, primarily as the `quest_giver` dialogue. An entry from the dataset from the game "Baldur's Gate II" is shown in fig. 4.1.


```
"tools": [
    "Korgan Bloodaxe and his fellows were on a mission to obtain the Book
    of Kaza for a book collector",
    "Korgan Bloodaxe and his fellows had a heated argument: only Korgan
    Bloodaxe remained alive afterwards"
],

"first_task_locations": [{
        "name": "",
        "description": "a tomb over in the lower crypts of the Graveyard
        District"
    }
],
"quest_giver": {
    "name": "Korgan Bloodaxe",
    "description": "a powerful warrior",
    "location": ""
},
"items": [{
    "name": "Book of Kaza",
    "description": "a grimoire",
    "amount": 1
}],
"characters": [{
    "name": "",
    "description": "book collector, a man",
    "location": ""
}],
```


(a) The plot points (**tools**) and the mentioned entities


```
"name": "Help Korgan Recover the Book of Kaza",
"objective": "bring the the Book of Kaza to the book collector",
"first_tasks": [
    "Head to a tomb over in the lower crypts of the Graveyard District",
    "find the Book of Kaza with Korgan Bloodaxe"
],

"description": "I'm scouring for a band of desperate adventurers to aid me
in a gallant task. Come and hear Korgan Bloodaxe's tale!\nA fortnight
past, me fellows and I were in the midst of obtaining an ancient text fer
our patron when a skirmish visited our midst. Vile words, alas, became a
lake of bloodshed. So it goes the sacred grimoire ne'er made it to our
benefactor's hand. And now he awaits its arrival with a zeal reserved for
a grog-blossom in an alekeg. The scuttlebutt is the pay is handsome and
worthy of note \u2014 sacks of loot and odd magics. The volume sits not
far from 'ere, and the bibliomaniac, he paces the floor in the meantime.
'Tis called the Book of Kaza.\nThe Book of Kaza is in some hobnail's tomb
over in the lower crypts of the Graveyard District. It's nae abandoned,
but rumors and half-truths only scare the young and infirm. Keeps the
curious away, me guess.\nWell... it'll take far more than a few shambling
bags of skin and stitches to deter Korgan Bloodaxe from a king's ransom! A
foolhardy jaunt into a hive of undead? How could ye resist?"
```


(b) The quest and the quest description

Figure 4.1: An entry from the video game quest dataset from the video game "Baldur's Gate II"

4.1.2 Constructing game world KG

To obtain the complete second fundamental element for the quest generation framework:

> **E2**: A *knowledge graph (KG)* which encapsulates the crucial entities from the game world, including the characters, locations and objects, along with the relations between these entities.

the extraction of relations between the entities provided with each quest is required. This section outlines the steps taken to complete the side-quest of this research - to construct the world KG for the games in the dataset.

In many cases, the data explicitly mentions the locations of the characters and items. This information is used to establish the character/object location relations in the KG - linking non-location entities to their respective locations with the "*located in*" relational label. Moreover, the description provided for each entity is also added to the KG.

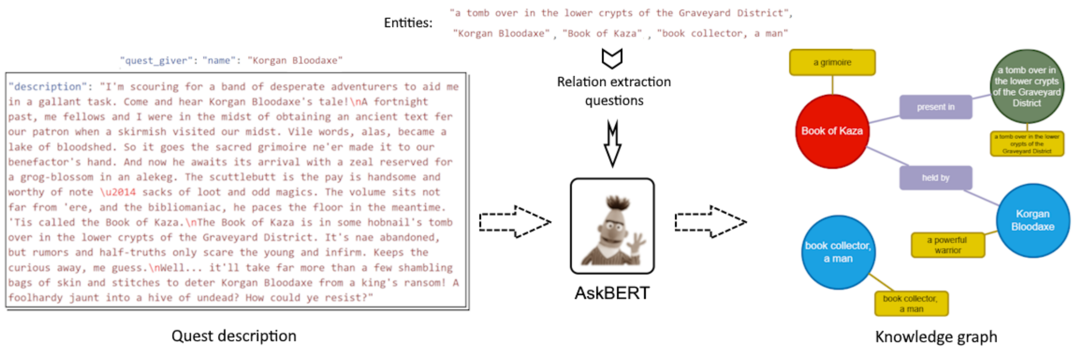


Figure 4.2: The knowledge graph construction process. For each quest, the quest giver name is appended to the quest description for the AskBERT model to extract relations for the mentioned entities

For the completion of the game world KG, the **AskBERT** KG extraction model by [8] is used. AskBERT is a question-answering large language model (LLM) designed to extract entities and their interrelations from a given narrative. This model is pre-trained on the SQuAD 2.0 dataset [65], enabling it to answer context-based questions or yield no response in the absence of relevant data. A key feature of AskBERT is the methodology used to answer queries in which it assesses the probability scores (logits) of each token in its vocabulary. It then identifies the indices of tokens that collectively formulate the answer within the input context. This input-agnostic characteristic of AskBERT makes it suited for extracting the entity relations to complete the game world KGs for this project.

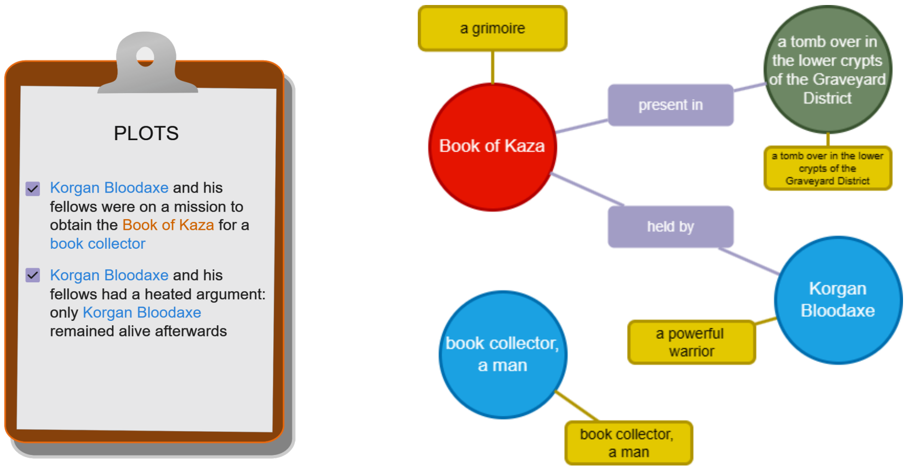
The AskBERT approach generates KG from a story in two steps: first, it extracts the locations, characters, and objects from the input story, and then it extracts the relations between these entities. At each step, the model is provided with the entire narrative in the input, followed by extractive questions to obtain the entities and relations. As the video game dataset by [51] already contains the entities, only the relation extraction capabilities of AskBERT are utilised for this project. The relations extracted as paired with the queries posed to the AskBERT model are as follows:

1. **connected to**: relation between locations. Queries of the format "*What location is next to {loc_a} in the story?*" are used to extract these relations, giving relations of the form (loc_a, *connected to*, loc_b).

2. **present in**: relation between a character or an object and a location. Questions of the format "*What is in {loc_a} in the story?*", "*Who is in {loc_a} in the story?*" and "*What location is {obj_a OR $char_a$} in the story?*" are used, giving relations in the form (obj_a, *present in*, loc_a) and ($char_a$, *present in*, loc_a).

3. **held by**: relation between a character and an object. The model is queried with "*Who has {obj_a} in the story?*", giving relations of the form (obj_a, *held by*, $char_a$).

For each quest entry in the dataset, the AskBERT model is provided with the quest `description` in the input text, followed by the extractive questions to obtain the relations between entities. The input is also prefixed with the quest giver's name before being input to the AskBERT model in the format "*I am <quest_giver_name>. <description>*". Without this context, the model defaults to the pronoun "I" as the answer in many instances because the quest descriptions are in the form of the `quest_giver` dialogue.



(a) The plot points (b) The knowledge graph

Figure 4.3: The story elements for the quest titled "*Help Korgan Recover the Book of Kaza*" from the game Baldur's Gate II. The entities in red are objects, those in blue are characters, and those in green are locations. The text in the yellow boxes is the description of the connected entity

After all the relations are extracted, a quest-level knowledge graph is obtained. The quest-level KG, together with the `plot_points`, form the story elements that make up the quest. Figure 4.3 shows the quest-level story elements - plot points and KG - obtained for the quest titled "*Help Korgan Recover the Book of Kaza*" from the game Baldur's Gate II. The quest level KGs for every quest from a particular game are combined to form the complete world KG of the game. This game world KG is constructed using the `networkX` package in python [66] and stored in the GML graph format. It is important to note that not every entry from the dataset results in the formation of a quest-level KG, and entries which do not form a KG are filtered out from the final dataset. The final processed dataset - after the removal of quests without `plot_points` and quests without KGs - comprises a total of 769 quests, which are then randomised and divided into fine-tuning and inference sets to be used by the quest generation model in a 90:10 ratio, allocating 692 quests for fine-tuning and 77 for inference.

In total, six world KGs - corresponding to the six games in the dataset are created. Table 4.2 enumerates the number of entities, relations, and the average node degree (the average number of relations per node) for each world KG generated. In alignment with the amount of data, "The Elder Scrolls: Skyrim" contributes the highest number of quests (268) and contains the highest number of entities, relations, and average node degrees, indicating a complex and interconnected KG. In contrast, games like

Game	**Filtered count**	**Entities**	**Relations**	**Avg degree**
Baldur's Gate I	65	290	235	1.62
Baldur's Gate II	89	281	243	1.73
The Elder Scrolls: Oblivion	196	586	618	2.11
The Elder Scrolls: Skyrim	268	909	998	2.20
Torchlight II	51	209	215	2.06
Minecraft	100	210	161	1.53

Table 4.2: World knowledge graph statistics per game from the dataset

"Minecraft" and "Baldur's Gate I" contain fewer entities and relations, with lower average node degrees, suggesting more superficial KG structures.

4.2 Quest Generation Model

The third element for the quest generation framework is (**E3**) A *quest generation model* which will generate quests using the story *plot points* and the *knowledge graph*. For this purpose, Meta's Llama2 13B model [13], accessible through the Huggingface platform, along with the Huggingface transformers library [67] is used.

Llama 2 is a leading-edge family of large language models, pre-trained and published by Meta. The Llama 2 family encompasses models of varying parameter sizes - 7B, 13B, and 70B. These models are specifically designed for text completion tasks and are known for their impressive performance across multiple benchmarks, including commonsense reasoning, world knowledge, and reading comprehension. They are pre-trained on publicly available textual data and are available to use for free for academic purposes on request. These models excel in generating coherent and contextually appropriate token sequences based on the provided input. The 13B parameter version of Llama 2 is selected for its optimal balance between size and performance, making it particularly suitable for the project's requirements. It exhibits performance metrics that are very close to those of its larger counterpart, the 70B model, while maintaining a significantly more manageable size.

The Llama 2 13B model was fine-tuned to generate quests from other elements of the framework which form the story components - (**E1**) the *plot points* and (**E2**) the *knowledge graph*. To enhance the efficiency of the model utilisation during fine-tuning and inference phases, the model parameter quantisation technique was adopted with parameters as mentioned in section 4.3.4). Along with quantisation, a low-rank adaptation (LoRA) of the parameters was also used. The combined strategy, called QLoRA [46], resulted in a substantial reduction in the model's memory requirements, an essential consideration for computational resource management. The final quest generation model was a Llama2-13B variant, loaded onto the memory with 4-bit quantisation and fine-tuned with low-rank adaptor parameters.

Being a text completion model, Llama2 generates new text that follows the given input sequence. Thus, the fine-tuning and inference phases of the quest generation model require the dataset compiled in section 4.1 to be in textual format, which the model would learn to complete by generating quests which follow the input information. The dataset contains three components: (1) the *plot points* for the quest, (2) the game's world *knowledge graph*, and (3) the quest itself. The process of combining these components into the input to the quest generation model is presented in the following section.

4.3 Experimental Setup

This section outlines the experimental setup designed to answer the sub-research questions of this project:

> **RQ1.a**: *How similar are the quests generated with varying amounts of contextual information provided by world knowledge graphs?*
>
> **RQ1.b**: *What effect does incorporating knowledge from knowledge graphs into LLMs have on improving the quality of generated quests in terms of story relatedness, coherence, contradiction and progression value?*

The quest generation model described in section 4.2 is fine-tuned on the data processed in section 4.1, to generate quests from the essential story elements - the plot points and the knowledge graph, with varying amounts of information from the obtained knowledge graphs. Section 4.3.1 describes the process of integrating the information from the data into a structured model input prompt. Section 4.3.2 outlines the methodology used to control the level of contextual information that was passed to the model to generate quests.

4.3.1 Generating knowledge grounded input prompts

The processed dataset contains two parts of the input: (1) the list of plot points in textual format and (2) the knowledge graph of the game world as a graph. The plot points can be directly added to the input prompt to the model as they are in textual format. However, the knowledge graph contains information in a non-textual format and requires further processing to be integrated into the model's input prompts.

As outlined in section 3.2, the existing methods that allow for the integration of graph knowledge into language models typically require access to the model's internal architecture to attach modules that process graph data. Such integration not only requires the pre-training of modules to process graph data but also entails the fine-tuning of the language model to assimilate the processed graph data. However, access to the underlying code and framework of large-scale language models may not always be feasible. Furthermore, this approach demands extensive pre-training and fine-tuning with substantial amounts of combined knowledge graph and text data, which exceeds the data resources available for this project.

As the underlying Llama2 model generates text from an input text sequence [13], the story elements - plot points and knowledge graph - obtained in section 4.1 need to be added to the model input prompt in a textual format. This process of appending contextual information with the input prompt to the model is known as *Knowledge grounding*, and it has been shown to decrease hallucinations, resulting in the model producing more coherent text with respect to the input [33, 35, 68].

Knowledge grounding input text does not require inherent changes to the model architecture and, in combination with fine-tuning methods, can result in a capable LLM for the task at hand. Consequently, for this project, the quest generation model was fine-tuned using knowledge-grounded input text. To create knowledge-grounded input prompts for the model fine-tuning, the processed dataset is converted in textual format. This conversion was crucial for creating model fine-tuning prompts that effectively integrate the story elements and quests into a cohesive text sequence.

The knowledge graph is stored as a graph containing the entities and relations between them which needs to be converted into textual descriptions. The list of all relations of an entity is created which each item represented as a (`entity`, `relation`, `entity`) triplet. These triplets are then converted into a textual description using the format described in table 4.3. For each entity, first, the entity description is mentioned, followed by its type. Then, every relation of the entity is written in textual format. An example of this conversion from the earlier mentioned quest is shown in fig. 4.4

Description	**Type**	**Relations**
{$\texttt{ent}_a$} is {$\texttt{ent}_{a_{desc}}$}	{$\texttt{ent}_a$} is a {$\texttt{ent}_{a_{type}}$}	[{$\texttt{ent}_a$} is {$\texttt{rel}_{(a,b)}$} {$\texttt{ent}_b$}, $\cdots$]

Table 4.3: Sentence format used to convert KG into text

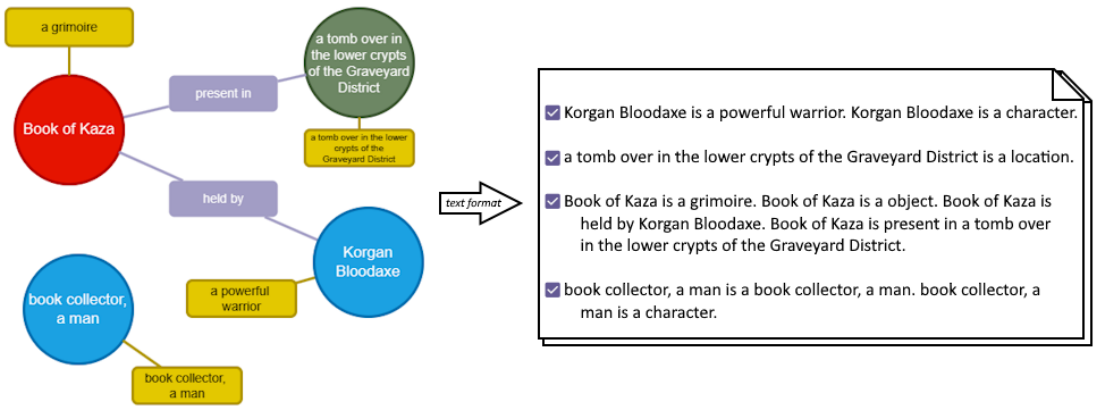


Figure 4.4: Converting knowledge graph to textual descriptions

Next, to combine the story elements - the plot points and the knowledge graph - along with the quest that follows, a simple input text prompt format was devised. In this format, the three parts of the prompt were delimited using three distinct tags, and the prompt was ended using an end tag. The "`### Background:`" tag was used to indicate the start of the KG data. The "`### Plots:`" tag denoted the beginning of the list of plot points related to the quest. After this, the quest generation prompt was added: "**The quest related to the above information is as follows:**", followed by the quest initiation tag "`### Quest:`". The quest was then added, and the prompt was ended with the "`### End`" tag.

The separation tags are placed with the aim of guiding the model into using the information provided in the tagged sections to generate quests. For fine-tuning, the quest generation model was trained to complete the text sequence. For inference, the user input prompt would end with the quest generation prompt, making the model generate the quest starting with the "`### Quest:`" tag and ending with the "`### End`" tag. The end tag allows the model to recognise the completion of the text sequence and thereby prevents the generation of extra tokens during inference.

4.3.2 Varying the amount of grounded information

The first sub-research question:

> **RQ1.a**: *How similar are the quests generated with varying amounts of contextual information provided by world knowledge graphs?*

aims to assess the impact of incorporating varying amounts of knowledge from the knowledge graph in quest generation from story plot points. As discussed in the previous section 4.3.1, the input to the model is grounded with knowledge to contain the plot points and information of the knowledge graph. To assess the effect of varying amounts of information from the knowledge graph, a depth parameter, `KG_DEPTH`, was used to control the amount of knowledge grounded in input by the quest generation model. The parameter values are as follows:

- At the foundational level `KG_DEPTH` = 0, no KG content was included, indicating that the model does not receive any information from the knowledge graph and would generate quests from the provided plot points alone.
- At depth `KG_DEPTH` = 1, the included information comprised only those entities directly mentioned in the plot points. The smaller KG extracted at this depth is the same quest level KG obtained in section 4.1.2.
- As the depth increases, `KG_DEPTH` ≥ 2, additional entities connected to those mentioned are incorporated, sourced from the game world KGs. Generally, at `KG_DEPTH` = n, all entities within $n-1$ hops from the entities mentioned in the plot points are included in the input text.

This mechanism acts as a controllable information extraction system, where the required information is extracted from the game world KG based on the provided story plot points.

4.3.3 Establishing the baseline model

The aim of the second research question:

> **RQ1.b**: *What effect does incorporating knowledge from knowledge graphs into LLMs have on improving the quality of generated quests in terms of story relatedness, coherence, contradiction and progression value?*

is to measure the impact of the integration of data from the knowledge graph on quest generation. To this end, the `KG_DEPTH` parameter is used to distinguish the quests generated with and without the integration of the knowledge graph. At `KG_DEPTH` = 0, the model is provided with no information from the knowledge graph and has to generate quests from the plot points alone. This forms as the *baseline model*. At `KG_DEPTH` = 1, the model receives information from the knowledge graph for the entities mentioned in the plot points. The inclusion of knowledge graph data for generation forms the second model, the *KG2Text model*. In line with the distinct *baseline* and *KG2Text* models, the input prompts for the models are shown in fig. 4.5. Figure 4.6 shows the working of these models. The difference between the inputs is the exclusion of KG data in the baseline model.

```
### Plots:
Korgan Bloodaxe and his fellows were on a mission to obtain the Book of Kaza for a book collector
Korgan Bloodaxe and his fellows had a heated argument: only Korgan Bloodaxe remained alive afterwards

The quest related to the above information is as follows:

### Quest:
Title: Help korgan recover the book of kaza
Objective: Bring the the book of kaza to the book collector
Tasks:
 Head to a tomb over in the lower crypts of the graveyard district.
 Find the book of kaza with korgan bloodaxe


### End
```

(a) Prompt used for the *baseline model*

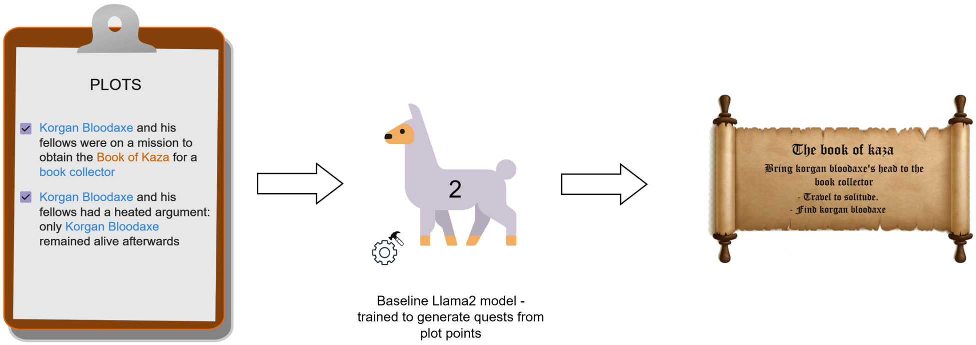
```
### Background:
Korgan Bloodaxe is a character. Korgan Bloodaxe is a powerful warrior.
a tomb over in the lower crypts of the Graveyard District is a location.
Book of Kaza is a object. Book of Kaza is a grimoire. Book of Kaza is held by Korgan Bloodaxe.
Book of Kaza is present in a tomb over in the lower crypts of the Graveyard District.


### Plots:
Korgan Bloodaxe and his fellows were on a mission to obtain the Book of Kaza for a book collector
Korgan Bloodaxe and his fellows had a heated argument: only Korgan Bloodaxe remained alive afterwards

The quest related to the above information is as follows:

### Quest:
Title: Help korgan recover the book of kaza
Objective: Bring the the book of kaza to the book collector
Tasks:
 Head to a tomb over in the lower crypts of the graveyard district.
 Find the book of kaza with korgan bloodaxe


### End
```

(b) Prompt used for *KG2Text model* at `KG_DEPTH` = 1

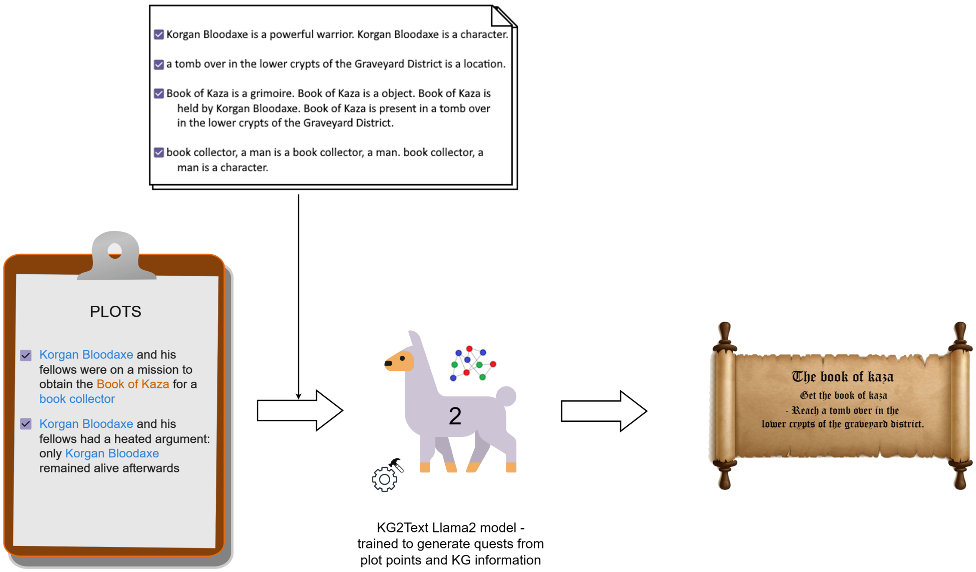
Figure 4.5: The prompts constructed with separation tags, defined in Section 4.3.1. The fine-tuning prompts include the complete text shown, where as the prompts for inference post fine-tuning include text until the generation prompt "**The quest related to the above information is as follows:**"

4.3.4 Fine-tuning and generating quests

Sections 4.3.1 to 4.3.3 outline the experiment design of the quest generation framework, which aligns with the goal of this project. The two distinct models, the *baseline model* and the *KG2Text model*, have been fine-tuned in an identical setup, except for the input text prompts. The *baseline model* is meant to generate quests from the story plot points alone. In contrast, the *KG2Text model* utilises the information from



(a) Baseline model



(b) KG2Text model

Figure 4.6: The distinct models part of the quest generation framework. The input to the *baseline model* shown in Figure 4.6a is the prompt designed in Figure 4.5a, whereas, the input to the *KG2Text model* shown in Figure 4.6b is the prompt designed in Figure 4.5b

the knowledge graph, all encompassed within the input prompts.

QLoRA configuration

The quest generation framework's underlying Llama2 model is fine-tuned with a combination of parameter-efficient fine-tuning (PEFT) methods, as mentioned in section 4.2. Firstly, the model parameters are loaded with a 4-bit precision quantisation technique to reduce the model's memory footprint significantly. Along with quantisation, the model parameters were also loaded with low-rank adaptation (QLoRA).

The following parameters were set as the QLoRA configuration:

- r The rank of the decomposed update matrices. This parameter was set to 64 after a trial and error loading where anything lower would cause the model

- α The scaling factor for the weight matrices. This parameter was set at 16 as [41] advice fixing this value to 16 for the best results instead of treating it as a hyper-parameter.

- `dropout` The dropout probability of the LoRA layers is used to reduce over-fitting by randomly selecting neurons to be ignored. This parameter was set to 0.1 after running a few fine-tuning experiments and observing that other values cause output repetition.

- `task_type` The type of fine-tuning or inference type being performed. As for this project, the task type was set to causal language modelling `CAUSAL_LM` - generating text sequences from a provided input sequence.

- `target_modules` The model modules which are targeted for decomposition. [41] suggest targeting the linear modules of the language model for the `CAUSAL_LM` task. For this project, all the linear modules of Llama2 were targeted.

With the above specified LoRA configuration and 4-bit quantisation, the Llama2 13B model takes approximately 12 GB of VRAM on an Nvidia Geforce RTX 4080 graphics card. The 4-bit quantisation technique involves pruning during the quantisation process [46], reducing the total number of parameters on the Llama2 13B model to $6,922,327,040$. Further on, with low-rank adaptation, the trainable parameters are reduced down to 3.62%, at $250,347,520$ parameters. The script to load the model with the mentioned QLoRA configuration was adopted from the official QLoRA repository from [46].

Fine-tuning setup

The Huggingface transformers library [67] provides the `transformers.Trainer` library for training and fine-tuning transformer-based models, such as Llama2. This library was used for fine-tuning the quest generation model, along with a modified version of the Llama2 fine-tuning script from the official @facebookresearch's Llama fine-tuning guide. The default optimiser settings from this script were used, i.e., Adam, with the initial learning rate of 2×10^{-4}. The batch size was set to 1 because higher values resulted in out-of-memory exceptions with 16GB of VRAM. The fine-tuning was set to run for 3 epochs over the training data (200 training steps, each step representing fine-tuning with one batch of data), with the gradient accumulation being done every four training steps. The default `EarlyStoppingCallback` was put in place to avoid over-fitting.

The fine-tuning goal for the quest generation models was to generate quests using the plot points with (and without, for the *baseline model*) the world data from the game knowledge graph. The fine-tuning prompts contained the actual quests as reference for the model to achieve. Llama2 uses the *CrossEntropyLoss* function for fine-tuning and parameter updating [13]. The fine-tuning ran for 50 steps in approximately 25 minutes on the RTX 4080 GPU for both *baseline* and *KG2Text* models, after which the early stopping mechanism was automatically activated.

Chapter 5

Evaluation

This chapter describes the final stages in answering the main research question:

> **RQ1**: *How does the integration of world knowledge affect the quality of quest generation using LLMs from narrative plot points?*

and the sub-research questions:

RQ1.a: *How similar are the quests generated with varying amounts of contextual information provided by world knowledge graphs?*

RQ1.b: *What effect does incorporating knowledge from knowledge graphs into LLMs have on improving the quality of generated quests in terms of story relatedness, coherence, contradiction and progression value?*

The quest generation capabilities of the Llama 2 models were examined under the lens of a creative language generation system, using ratings provided by human evaluators. The quest generation framework specified in chapter 4 is evaluated in an experimental setting to determine the impact of incorporating (varying amounts of) KG data into the framework. Section 5.1 outlines the experimental setup designed to answer **RQ1.a**. Section 5.2 outline the design and results of the evaluation methodology employed to answer **RQ1.b**. Finally, section 5.3 discusses the reported results.

5.1 Impact of varying amounts of KG data

This section outlines the evaluative setup designed to approach the first sub-research question:

> **RQ1.a**: *How similar are the quests generated with varying amounts of contextual information provided by world knowledge graphs?*

The setup involved modifying the amount of background information from the knowledge graphs included in the input prompts for quest generation, thereby creating varying contexts for quest generation, using the `KG_DEPTH` parameter, as detailed in section 4.3.2. The resulting quests were subsequently assessed using the `BERTScore` text similarity metric [63]. `BERTScore` computes text sequence similarity utilising sentence representations derived from the pre-trained BERT model. [63]

have observed that BERTScore exhibits a higher correlation with human evaluations and demonstrates greater resilience in handling complex examples compared to n-gram-based metrics. BERTScore yields an F1 score when comparing two text fragments, with a score of 0 indicating no similarity and 1 indicating identical texts.

The dataset described and processed in section 4.1 contains the actual quest from the game, including details from the fields title, objective and first_tasks. This quest information is treated as the gold standard for evaluation purposes, representing the ideal output that an optimal quest generation model would yield. Consequently, the quests generated by the model are benchmarked against these actual quests to assess the fidelity of the generated content to the actual quests. Additionally, the generated quests undergo an intra-comparative analysis to evaluate the effects of incorporating different levels of knowledge graph data. This systematic variation and comparative analysis provides insights to understand how different amounts of world information influence the resulting outputs. Table 5.1 illustrates the comparative analysis of this similarity.

		actual quests	*baseline*	*KG2Text*	
Model type	KG_DEPTH	-	0	1	2
baseline	0	0.71 ± 0.05			
KG2Text	1	0.74 ± 0.07	0.75 ± 0.06		
	2	0.73 ± 0.06	0.73 ± 0.06	0.77 ± 0.07	
	3	0.68 ± 0.09	0.69 ± 0.09	0.70 ± 0.09	0.70 ± 0.09

Table 5.1: Average BERTScore ($\mu \pm \sigma$) obtained on comparison of generated quests with varying KG_DEPTH and actual quests.

Quests generated without KG data average a score of 0.71 ± 0.05 when compared with actual quests. With the inclusion of KG data at KG_DEPTH = 1, the similarity to actual quests increases slightly to 0.74 ± 0.07. When compared to the baseline model, the score is 0.75 ± 0.06. Increasing the KG_DEPTH to 2, the generated quests seem to become slightly dissimilar compared to the actual quests, giving a score of 0.73 ± 0.06. These quests are more similar to the quests generated at KG_DEPTH = 1 with a score 0.77 ± 0.07. At KG_DEPTH 3, the quests seem to stray away from the actual quests, giving the least similar quests with a score of 0.68 ± 0.09.

5.2 AI Quest Generation Survey

This section outlines the evaluative setup to answer the second sub-research question:

> **RQ1.b**: *What effect does incorporating knowledge from knowledge graphs into LLMs have on improving the quality of generated quests in terms of story relatedness, coherence, contradiction and progression value?*

To this end, an online survey titled "AI Quest Generation Survey" was designed to enable human participants to evaluate the quests generated by the model in relation to the original narrative. The quality of the quest generation framework is measured by asking the participants to judge the generated quests on four assessment criteria. The assessment criteria focused on aspects such as relatedness to the story, contradiction and coherence with the story, and narrative progression value.

The hypotheses to be tested with the survey are stated as follows:

H0 *Null hypothesis*: The quests generated by including KG data along with plot points are the same as those generated without KG data.

H1 *Alternate hypothesis*: The quests generated by including KG data along with plot points are better (in quality) than those generated without KG data.

5.2.1 Survey design

The survey was designed to require approximately 25-30 minutes for completion, and it was deployed using the Microsoft Forms platform. It targeted individuals with a proficient understanding of English and familiarity with video game quests. The survey commenced with an introductory page outlining the estimated completion time, eligibility criteria, and a privacy statement - no personal information was collected during the survey. The Research Ethics Committee of the University of Twente has approved the procedure outlined in this section.

Participants were first requested to self-assess their familiarity with video game quests and proficiency in English on a numerical scale that ranged from 1 (indicating limited knowledge) to 5 (denoting expert understanding). The responses of participants who indicated either or both proficiency below three were rejected. This rating scale was followed by a concise overview of the question format to acquaint participants with the survey's structure. The main body of the survey comprised ten questions, each presenting the quest background story from the `description` field from the dataset. The story was followed by two generated quests - one from the *baseline model* and the other from the *KG2Text model* with `KG_DEPTH = 1`. This `KG_DEPTH` was selected as it was the most similar to the actual quests. These quests were randomised for each question to ensure unbiased responses. The participants were not informed about the quests being generated using an AI model until they completed the survey.

Participants were first instructed to select the quest they deemed more appropriate for the given story. Subsequently, they were asked to rate each quest based on the following criteria, aligning with the objectives of Research Question **RQ1.b**:

- **Relatedness**: *"The quest is related to the given story"* - Evaluating if the quest is relevant to the provided story.
- **Contradiction**: *"The quest contains contradictory statements with respect to the story shown"* - Determining if the quest contains statements that contradict the story.
- **Coherence**: *"The quest follows the shown storyline consistently"* - Assessing the consistency of the quest with the storyline.
- **Value**: *"The quest contributes to progressing further in the given story"* - Judging the quest's alignment to the story plot progression.

These criteria were essential for understanding the influence of incorporating knowledge graphs (KGs) into the quest generation process. While relatedness, coherence, and value focused on the quest's alignment with story progression, contradiction assessed the language generation model's affinity towards generating inconsistent content or hallucinating.

Participants rated the quests using a 5-point Likert scale, expressing their agreement or disagreement with each criterion, ranging from "Strongly Disagree" (1) to "Strongly Agree" (5). Recruitment of survey participants was conducted through various video game-related sub-communities on platforms like Reddit and Discord, including the r/SampleSize subreddit, which is dedicated to scientific studies and surveys.

For the survey, quest pairs were selected from the pool of 77 quests in the inference set, as mentioned in section 4.1, generated by the *baseline* and *KG2Text* models. The individual quest comparison `BERTScore` between the *baseline* outputs and *KG2Text* outputs at `KG_DEPTH` = 1 obtained in section 5.1 was used to pick the quests for the survey. The generated quest pairs were ordered according to their BERTScore values, with a higher score indicating greater similarity. The ordered list was then divided into eight segments, and a quest pair was randomly selected from each segment, giving 10 quest pairs. The selected quests are shown in appendix A. This approach ensured a diverse representation of quest pairs based on similarity, and randomised in terms of the inclusion or exclusion of KGs per question.

5.2.2 Survey Results

A total of 26 responses were collected from the survey. However, four of these responses were excluded from the final analysis because two responses were completed in under 5 minutes, and two responses exhibited a polarised rating pattern (exclusively rating 5 - Strongly Agree or 1 - Strongly Disagree for all questions), one of which also self-reported a familiarity level with video games as 1. Consequently, the analysis was conducted on the remaining 22 valid responses.

Within this refined response set, participants, on average, rated their English proficiency at 4.68 and their familiarity with video game quests at 4.23. The average duration taken by participants to complete the survey was 31 minutes and 54 seconds, slightly exceeding the estimated time.

The combined evaluations from these participants yielded a total of 220 ratings across the specified criteria for quests generated both with and without the inclusion of KG data. Out of these 220 assessments, quests that incorporated KG were favoured 146 times, significantly outnumbering the non-KG quests, which were preferred only 37 times. In the remaining 37 instances, participants perceived the quests as being very similar, indicating no distinct preference. This data, also visualised in fig. 5.1, conveys that the quests generated with the integration of KG data are highly preferred as compared to those generated without KG data. Figure 5.2 provide a visual representation of the distribution of agreement levels for the evaluation criteria, as gauged on the 5-point Likert scale.

The results reveal an apparent tendency for quests generated without the inclusion of knowledge graph (KG) data to exhibit a higher degree of perceived contradiction with the story narrative compared to those generated with KG data. Conversely, quests incorporating KG data are observed to receive more favourable assessments

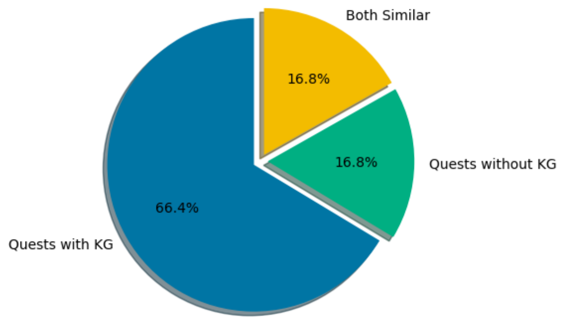


Figure 5.1: Survey respondents' preferences for *baseline* quests and *KG2Text* quests

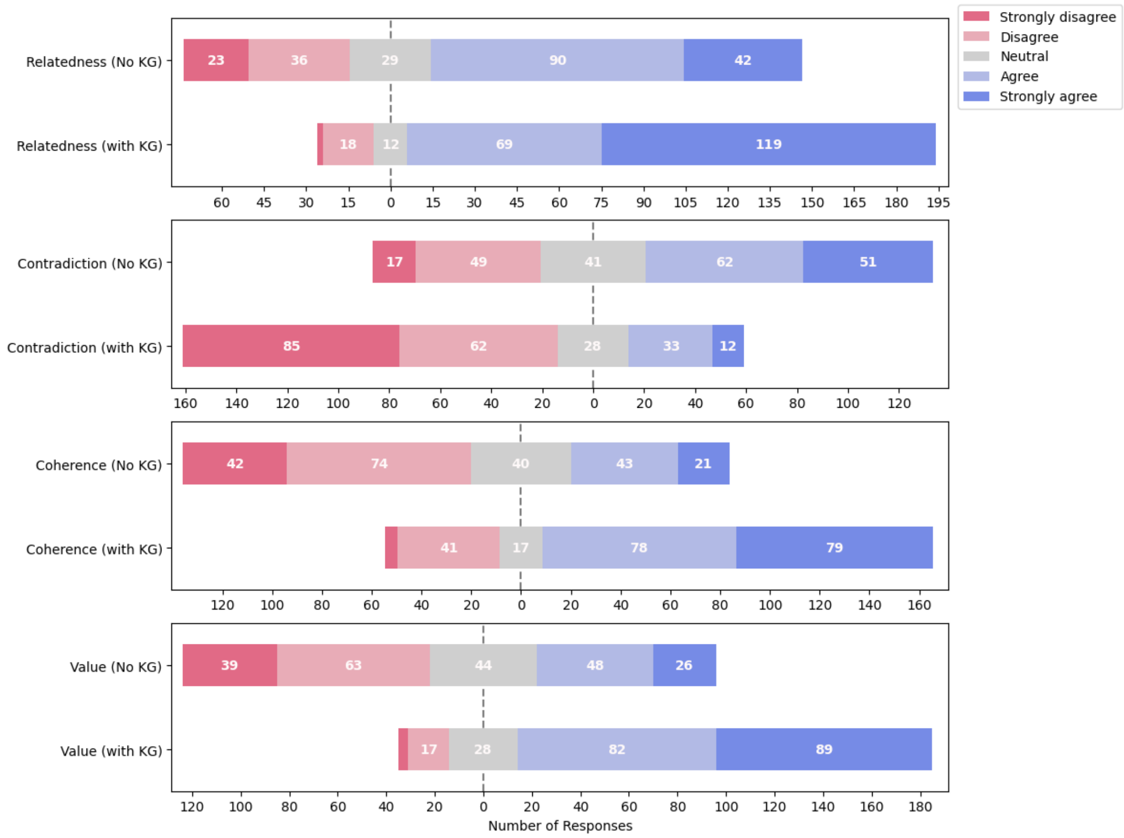


Figure 5.2: Likert scale data distribution per criterion for the generated quests. For each criterion, the first plot shows the response distribution for quests generated without KG data and the second plots shows the same for quests generated with KG data.

in terms of relatedness, coherence, and value.

Since rating data typically exhibits a high degree of interrelation, a statistical significance test was conducted with the ratings obtained in the survey to ascertain the statistical significance of the results. For this analysis, the Likert scale was treated as ordinal values, assigning (1) for "Strongly Disagree" and (5) for "Strongly Agree".

Considering the relatively small sample size and the nature of the evaluated outputs (i.e., generated quests) not adhering to a normal distribution, a non-parametric approach was requisite. The two-sided Wilcoxon signed-rank test was identified as the most suitable method for this analysis, as the measurement variables (rating criteria) are on an ordinal scale and are measured under two dependent categories (quests generated with and without KG data).

The criterion to reject the null hypothesis

> **H0**: The quests generated by including KG data along with plot points are the same as those generated without KG data.

set to $p<0.05$, indicating that the rating distributions for the two models are different with statistical significance. The results, as detailed in table 5.2, demonstrate that the rating distribution for each criterion is significantly different (as indicated in the table with † denoting significance at $p \ll 0.05$), with the quests generated with the inclusion of KG data performing much better than those generated without the KG data. This finding allows for the acceptance of the primary hypothesis with a high level of statistical confidence, affirming the improvement in the quality of quests generated by integrating KG data.

The observation from the Likert rating distribution from fig. 5.2 is corroborated by the data presented in table 5.2, which indicates that, on average, quests generated with KG data are rated higher for relatedness (4.29 vs. 3.42), coherence (3.84 vs. 2.67), and value (4.07 vs. 2.81). They are also perceived as less contradictory compared to quests generated without KG data (2.20 vs. 3.37). The standard deviations show relatively similar variability in responses across both types of quests.

Criteria	**mean (μ)**		**std. dev (σ)**		**median**	
	No KG	*KG*	*No KG*	*KG*	*No KG*	*KG*
Relatedness	3.42	4.29 †	1.26	0.96	4	5
Contradiction	3.37	2.20 †	1.27	1.25	4	2
Coherence	2.67	3.84 †	1.25	1.17	2	4
Value	2.81	4.07 †	1.29	1.00	3	4

Table 5.2: Mean, standard deviation, median and p-values for the evaluated criteria. Scores under "No KG" refer to the ratings of the *baseline model*, whereas "KG" refers to that of the *KG2Text model*. † indicates a significant difference ($p \ll 0.05$)

5.3 Discussions

Sections 5.1 and 5.2 presents the results of evaluating the quest generation framework of this project. The comprehensive evaluation of the quest generation models, both the *baseline* and *KG2Text* models, offers insightful observations on the impact of knowledge graph (KG) integration on quest generation quality. The evaluation system first examined how similar the generated quests are as compared to the actual quests as well as with varying amounts of data from the KG (using `KG_DEPTH`).

5.3.1 Findings

To answer the first sub-research question:

> **RQ1.a**: *How similar are the quests generated with varying amounts of contextual information provided by world knowledge graphs?*

similarity scores using the `BERTScore` metric were calculated. The study investigated the impact of varying levels of KG data on quest generation from plot points. The scores obtained demonstrated that quests generated by incorporating data about the entities directly mentioned in the plot points (`KG_DEPTH` = 1) were the most similar to the actual quests compared to other generated outputs. Moreover, increasing the `KG_DEPTH` reduces this similarity. At the highest level tested, (`KG_DEPTH` = 3), the generated quests are the most dissimilar to the actual quests, as well as quests generated at lower levels.

The fine-tuning goal of the quest generation model was to generate quests like the actual quests, using story plot points and world data from the knowledge graph. The similarity scores obtained showcase that incorporating information at `KG_DEPTH` = 1 produces the most real-world-like quests, beyond which the effectiveness in mimicking actual quests decreases. The reduction in scores could be attributed to potential over-fitting or excessive complexity introduced at higher KG depths. Moreover, increasing the amount of KG data as textual prompts to the LLM uses up valuable input tokens for potentially unnecessary information.

In addressing the second research question:

> **RQ1.b**: *What effect does incorporating knowledge from knowledge graphs into LLMs have on improving the quality of generated quests in terms of story relatedness, coherence, contradiction and progression value?*

the survey results reveal a clear preference for quests generated with the inclusion of KG data, as evidenced by higher ratings in relatedness, coherence, and value and lower ratings in contradiction. The results suggest that the inclusion of KG data significantly enhances the overall quality of the quests, making them more aligned with the story, coherent, and valuable in terms of story progression. The statistical analysis further substantiates these findings, indicating a statistically significant improvement in quest quality with the inclusion of KG data. The survey results, along with the `BERTScore` values, reveal that the quests generated by including world knowledge from the game knowledge graph at `KG_DEPTH` = 1 are more aligned with the actual quests, being the most similar to the actual quests from all the generated quests.

In summary, the evaluation shows that the inclusion of KG data (up to `KG_DEPTH = 2`) results in generated stories being more like the actual quests found in games. The survey results highlight that the inclusion of KG data in quest generation from story plot points improves the perceived quality of the quests. The quests generated with KG data are perceived to be more related and coherent with the background narrative and contribute better in terms of narrative progression value. Moreover, the quests generated with the inclusion of KG data appear to have significantly fewer contradictory statements with respect to the narrative.

Chapter 6

Discussions and Conclusions

This chapter concludes the research work done in this thesis. The research methodology and the findings are presented in section 6.1. After that, the limitations of the work are laid out in section 6.2. The major contributions and value additions of the thesis are discussed in section 6.3. Finally, potential extensions to the research are suggested in section 6.4.

6.1 Findings

Circling back to chapter 1, this study embarked on the quest to answer the main research question:

> **RQ1**: *How does the integration of world knowledge affect the quality of quest generation using LLMs from narrative plot points?*

To better answer the main research question, it was divided into two sub-research questions:

RQ1.a: *How similar are the quests generated with varying amounts of contextual information provided by world knowledge graphs?*

RQ1.b: *What effect does incorporating knowledge from knowledge graphs into LLMs have on improving the quality of generated quests in terms of story relatedness, coherence, contradiction and progression value?*

A literature study was done to understand the current state-of-the-art methodologies in video game content generation, including generating quests. The lack of studies on automated quest generation using LLMs and story plots was highlighted, and the importance of incorporating world information from knowledge graphs was identified, as outlined in chapter 3. Then, a quest generation framework was established, which included three fundamental elements:

- **E1** A defined set of *plot points* which together form up the game's narrative;
- **E2** A *knowledge graph (KG)* which encapsulates the crucial entities from the game world, including the characters, locations and objects, along with the relations between these entities; and

E3 A *quest generation model* which will generate quests from the story components - the *plot points* and the *knowledge graph*.

The dataset by [51] fulfilled these requirements to the highest extent and was further processed to obtain the necessary elements. Next, an experimental design was constructed to fine-tune the *quest generation model* to produce quests from story plot points and world knowledge, as presented in chapter 4.

The similarity of the generated quests with the actual quests and amongst themselves was analysed using the `BERTScore` similarity metric, and the impact of incorporating knowledge graphs on the quality of generated quests was assessed using an online survey, both the processes described in detail in chapter 5. The findings of the survey indicate that quests created with KG data are significantly better in terms of relatedness to the story, coherence, and progression value while also exhibiting fewer contradictions when compared to quests generated without KG data. This improvement in quality was also statistically validated and suggested that incorporating contextual information from KGs effectively enhances the narrative structure and relevance of the quests.

6.2 Limitations

The research work in this thesis has some limitations, as described next. Firstly, the `BERTScore` values obtained in section 5.1 do not convey information about the quality of generated quests at `KG_DEPTH` 2 and 3. Although the quests seem to be less similar to the actual quests, there is no indication of these quests being better or worse than the *baseline* or actual quests. That being said, the actual quests from the games are considered as the ground-truth data for generation, and the quest generation framework aims to generate quests as close to the actual quests as possible. Therefore, setting `KG_DEPTH` to 1 does not indicate the best quality quests among the generated ones, but rather, the most similar to the actual quests.

Another area for improvement of this research is the method of incorporating world knowledge data. The knowledge graph data was integrated into the quest generation models in a textual format. Such textual representations do not capture the structured nature of the data. Employing graph representations alongside language models in a multi-modal framework can potentially enhance the quality of the generated quests and reduce the number of input tokens required for the model. Although this approach would necessitate extensive model training, it represents a promising direction for future advancements in game development by utilising the methods outlined in section 3.2. However, the scope of this project was to use readily available pre-trained LLMs where access to the model architecture and source code is only sometimes feasible.

6.3 Contributions

The contributions of this research are two-fold:

1. Firstly, a systematic and reproducible quest generation framework is provided. As described in chapter 4, this framework is capable of generating quests using three fundamental elements: the narrative *plot points*, controllable levels of

world knowledge from *knowledge graphs*, and a *quest generation model*, which is a state-of-the-art LLM, fine-tuned to generate quests using the previous elements.

2. Secondly, the dataset provided by [51] is now augmented with a complete world knowledge graph for each of the games in the dataset. These knowledge graphs are extracted using the quest descriptions and the identified entities, as outlined in section 4.1.

6.4 Future work

This research could be extended in several ways, a few of which are elaborated next.

Firstly, the quest generation framework could be extended to automatically extract story plots and the world knowledge graph from a given narrative. In its current state, the quest generation framework can generate individual quests from an existing set of plot points and world knowledge. The `AskBERT` model [53] can be used in its total capacity to extract the world knowledge graph from a story. Current research on event extraction from text [69, 70] makes the idea of an end-to-end story-to-quest generation system a feasible next step.

Secondly, a dynamic approach to recording the game world could be explored to create on-the-fly quest generation based on the current state of the narrative and world entities. The work on tracking the game world using graphs by [48] can be combined with the quest generation framework created in this project for this extension.

6.5 Conclusion

The aim of this thesis was to assess the impact of including world knowledge on the quality of quest generation using story plot points. After following a literature study, creating a quest generation framework, experimenting with controllable amounts of knowledge incorporation in quest generation and evaluating the results, it can be concluded that integrating world knowledge improves the quality of quests generated from narrative plot points. Including information about the entities mentioned directly in the plot points generate quests which are most similar to actual quests. The quests generated with the inclusion of world knowledge are perceived to be more related and coherent with the background story plot and contribute more to the story progression value. Moreover, the quests generated without the inclusion of world knowledge tend to contain more contradictory statements. Potential improvements to the quest generation framework are also suggested, which could help game developers streamline the processes of creating quests that stay true to the overarching game narrative.

Bibliography

[1] J. Clement. Topic: COVID-19 impact on the gaming industry worldwide, April 2023. [Online; accessed 13. Apr. 2023].

[2] Chloi Rad. 11 Games You Didn't Know Were Based on Books - IGN. *IGN*, May 2015.

[3] Quinn Levandoski. 10 Best Video Games Based On Books. *ScreenRant*, August 2021.

[4] Jeremiah McCall. *Gaming the Past:Using Video Games to Teach Secondary History*. Taylor & Francis, Andover, England, UK, November 2022.

[5] Josh Sokol and Karissa Schaefer. Video games are the modern form of storytelling. *Berkeley Beacon*, January 2021.

[6] Rebekah Valentine. Turns Out The Hardest Part of Making a Game Is...Everything - IGN. *IGN*, October 2022.

[7] Arwa I. Alhussain and Aqil M. Azmi. Automatic Story Generation: A Survey of Approaches. *ACM Comput. Surv.*, 54(5):1–38, May 2021.

[8] Prithviraj Ammanabrolu, Wesley Cheung, Dan Tu, William Broniec, and Mark Riedl. Bringing Stories Alive: Generating Interactive Fiction Worlds. *AIIDE*, 16(1):3–9, October 2020.

[9] Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. K-BERT: Enabling Language Representation with Knowledge Graph. *AAAI*, 34(03):2901–2908, April 2020.

[10] Ye Liu, Yao Wan, Lifang He, Hao Peng, and Philip S. Yu. KG-BART: Knowledge Graph-Augmented BART for Generative Commonsense Reasoning. *AAAI*, 35(7):6418–6425, May 2021.

[11] Phillip Schneider, Tim Schopf, Juraj Vladika, Mikhail Galkin, Elena Simperl, and Florian Matthes. A decade of knowledge graphs in natural language processing: A survey. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 601–614, Online only, November 2022. Association for Computational Linguistics.

[12] Ben Kybartas and Clark Verbrugge. Analysis of ReGEN as a Graph-Rewriting System for Quest Generation. *IEEE Trans. Comput. Intell. AI Games*, 6(2):228–242, November 2013.

[13] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open Foundation and Fine-Tuned Chat Models. *arXiv*, July 2023.

[14] Chee Siang Ang. Rules, gameplay, and narratives in video games. *Simulation & Gaming*, 37(3):306–325, September 2006.

[15] Young Ho Roh. Quest systems in Role Playing Games - Ludogogy, April 2023. [Online; accessed 14. Apr. 2023].

[16] Jonathon Doran and Ian Parberry. A prototype quest generator based on a structural analysis of quests from four MMORPGs. In *PCGames '11: Proceedings of the 2nd International Workshop on Procedural Content Generation in Games*, pages 1–8. Association for Computing Machinery, New York, NY, USA, June 2011.

[17] Kristen K. Yu, Matthew Guzdial, and Nathan Sturtevant. The Definition-Context-Purpose Paradigm and Other Insights from Industry Professionals about the Definition of a Quest. *AIIDE*, 17(1):107–114, October 2021.

[18] Judith van Stegeren and Jakub Myśliwiec. Fine-tuning GPT-2 on annotated RPG quests for NPC dialogue generation. In *FDG '21: Proceedings of the 16th International Conference on the Foundations of Digital Games*, pages 1–8. Association for Computing Machinery, New York, NY, USA, August 2021.

[19] Judith van Stegeren. Flavor text generation for role-playing video games. *University of Twente Research Information*, March 2022.

[20] Hanqing Zhang, Haolin Song, Shaoyu Li, Ming Zhou, and Dawei Song. A survey of controllable text generation using transformer-based pre-trained language models. *ACM Comput. Surv.*, 56(3), oct 2023.

[21] Joseph Weizenbaum. ELIZA — a computer program for the study of natural language communication between man and machine. *Commun. ACM*, 26(1):23–28, January 1983.

[22] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[23] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics.

[24] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. *arXiv*, December 2013.

[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. *arXiv*, June 2017.

[26] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv*, September 2014.

[27] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[28] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. *arXiv*, May 2020.

[29] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[30] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1), jan 2020.

[31] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online, July 2020. Association for Computational Linguistics.

[32] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: Open and Efficient Foundation Language Models. *arXiv*, February 2023.

[33] Jie Huang and Kevin Chen-Chuan Chang. Towards reasoning in large language models: A survey. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1049–1065, Toronto, Canada, July 2023. Association for Computational Linguistics.

[34] Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. Are Emergent Abilities of Large Language Models a Mirage? *arXiv*, April 2023.

[35] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of Hallucination in Natural Language Generation. *ACM Comput. Surv.*, 55(12):1–38, March 2023.

[36] Robert Logan, Nelson F. Liu, Matthew E. Peters, Matt Gardner, and Sameer Singh. Barack's wife hillary: Using knowledge graphs for fact-aware language modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5962–5971, Florence, Italy, July 2019. Association for Computational Linguistics.

[37] Michael McCloskey and Neal J. Cohen. Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. In *Psychology of Learning and Motivation*, volume 24, pages 109–165. Academic Press, Cambridge, MA, USA, January 1989.

[38] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[39] Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 1950–1965. Curran Associates, Inc., 2022.

[40] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR, 09–15 Jun 2019.

[41] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. *arXiv*, June 2021.

[42] Tim Dettmers and Luke Zettlemoyer. The case for 4-bit precision: k-bit Inference Scaling Laws. *arXiv*, December 2022.

[43] Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. LLM.int8(): 8-bit Matrix Multiplication for Transformers at Scale. *arXiv*, August 2022.

[44] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers. *arXiv*, October 2022.

[45] Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. SmoothQuant: Accurate and efficient post-training quantization for large language models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 38087–38099. PMLR, 23–29 Jul 2023.

[46] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient Finetuning of Quantized LLMs. *arXiv*, May 2023.

[47] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia D'amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, and Antoine Zimmermann. Knowledge Graphs. *ACM Comput. Surv.*, 54(4):1–37, July 2021.

[48] Iwona Grabska-Gradzińska, Leszek Nowak, Wojciech Palacz, and Ewa Grabska. Application of graphs for story generation in video games. In *Proceedings of the 2021 Australasian Computer Science Week Multiconference*, ACSW '21, New York, NY, USA, 2021. Association for Computing Machinery.

[49] YunSeok Choi, SuAh Kim, and Jee-Hyong Lee. Recurrent Neural Network for Storytelling. In *2016 Joint 8th International Conference on Soft Computing and Intelligent Systems (SCIS) and 17th International Symposium on Advanced Intelligent Systems (ISIS)*, pages 841–845. IEEE, August 2016.

[50] Sayan Ghosh, Mathieu Chollet, Eugene Laksana, Louis-Philippe Morency, and Stefan Scherer. Affect-LM: A Neural Language Model for Customizable Affective Text Generation. *ACL Anthology*, pages 634–642, July 2017.

[51] Susanna Värtinen, Perttu Hämäläinen, and Christian Guckelsberger. Generating Role-Playing Game Quests With GPT Language Models. *IEEE Transactions on Games*, pages 1–12, December 2022.

[52] Suzan Al-Nassar, Anthonie Schaap, Michael Van Der Zwart, Mike Preuss, and Marcello A. Gómez-Maureira. QuestVille: Procedural Quest Generation Using NLP Models. In *FDG '23: Proceedings of the 18th International Conference on the Foundations of Digital Games*, pages 1–4. Association for Computing Machinery, New York, NY, USA, April 2023.

[53] Prithviraj Ammanabrolu, William Broniec, Alex Mueller, Jeremy Paul, and Mark Riedl. Toward automated quest generation in text-adventure games. In *Proceedings of the 4th Workshop on Computational Creativity in Language Generation*, pages 1–12, Tokyo, Japan, 29 October–3 November 2019. Association for Computational Linguistics.

[54] Wenhu Chen, Yu Su, Xifeng Yan, and William Yang Wang. KGPT: Knowledge-grounded pre-training for data-to-text generation. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8635–8648, Online, November 2020. Association for Computational Linguistics.

[55] Mausam Mausam. Open information extraction systems and downstream applications. In *IJCAI'16: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 4074–4077. AAAI Press, July 2016.

[56] Swarnadeep Saha and Mausam. Open information extraction from conjunctive sentences. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2288–2299, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics.

[57] Swarnadeep Saha, Harinder Pal, and Mausam. Bootstrapping for numerical open IE. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 317–323, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[58] Harinder Pal and Mausam. Demonyms and compound relational nouns in nominal open IE. In *Proceedings of the 5th Workshop on Automated Knowledge Base Construction*, pages 35–39, San Diego, CA, June 2016. Association for Computational Linguistics.

[59] Janara Christensen, Mausam, Stephen Soderland, and Oren Etzioni. An analysis of open information extraction based on semantic role labeling. In *K-CAP '11: Proceedings of the sixth international conference on Knowledge capture*, pages 113–120. Association for Computing Machinery, New York, NY, USA, June 2011.

[60] Chenguang Wang, Xiao Liu, and Dawn Song. Language models are open knowledge graphs. *CoRR*, abs/2010.11967, 2020.

[61] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. Language Models as Knowledge Bases? *ACL Anthology*, pages 2463–2473, November 2019.

[62] Benjamin Heinzerling and Kentaro Inui. Language Models as Knowledge Bases: On Entity Representations, Storage Capacity, and Paraphrased Queries. *ACL Anthology*, pages 1772–1791, April 2021.

[63] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. BERTScore: Evaluating Text Generation with BERT. *arXiv*, April 2019.

[64] Ananya B. Sai, Akash Kumar Mohankumar, and Mitesh M. Khapra. A Survey of Evaluation Metrics Used for NLG Systems. *ACM Comput. Surv.*, 55(2):1–39, January 2022.

[65] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[66] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using networkx. In Gaël Varoquaux, Travis Vaught, and Jarrod Millman, editors, *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, Pasadena, CA USA, 2008.

[67] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771, 2019.

[68] Wenhao Yu, Chenguang Zhu, Zaitang Li, Zhiting Hu, Qingyun Wang, Heng Ji, and Meng Jiang. A survey of knowledge-enhanced text generation. *ACM Comput. Surv.*, 54(11s), nov 2022.

[69] Xiao Liu, Heyan Huang, Ge Shi, and Bo Wang. Dynamic prefix-tuning for generative template-based event extraction. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5216–5228, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[70] Hong Chen, Raphael Shu, Hiroya Takamura, and Hideki Nakayama. GraphPlan: Story generation by planning with event graph. In Anya Belz, Angela Fan, Ehud Reiter, and Yaji Sripada, editors, *Proceedings of the 14th International Conference on Natural Language Generation*, pages 377–386, Aberdeen, Scotland, UK, August 2021. Association for Computational Linguistics.

Appendix A

Appendix A - Outputs Selected for the survey

Game: **The Elder Scrolls: Skyrim**

Description: **Isran:** "The day hasn't been won while the vampire Harkon still walks Tamriel.

Let me address the Dawnguard, and then we'll be off. The men deserve to know that we've finally gained the upper hand. It's time we take the fight to their door.

Gather your things and rest well, my friend. We meet outside Castle Volkihar."

Plot points:

- Isran will address the player's fellow Dawnguard members before leaving for Castle Volkihar: the Dawnguard has gained the upper hand in the fight against the vampires

KG2Text:

- Isran is a character. Isran is present in Tamriel.
- Tamriel is a location. Tamriel is connected to Castle Volkihar.
- Castle Volkihar is a location.
- Harkon is a character. Harkon is present in Castle Volkihar.

No KG output: **Title:** The vampire war

Description: Find isran

Tasks:

- Travel to castle volkihar.
- Find isran.

Text2KG output: **Title:** The vampire war

Description: Find harkon in castle volkihar

Tasks:

- Find your way to castle volkihar

Actual output: **Title:** Judgment

Description: Destroy harkon

Tasks:

- Head to castle volkihar.
- Meet up with isran and the rest of your dawnguard allies

Game: **The Elder Scrolls: Skyrim**

Description: **Alvor:** "What? A dragon? In Helgen? That explains what I saw earlier... a dragon flying down from the south... I was hoping I was wrong about what I thought it was... Didn't want to believe my own eyes, is all.

The Jarl needs to know if there's a dragon on the loose. Riverwood is defenseless... We need to get word to Jarl Balgruuf in Whiterun to send whatever soldiers he can. If you'll do that for me, I'll be in your debt."

Plot points:

- the player told Alvor that there was a dragon in Helgen

KG2Text:

- Alvor is a character. Alvor is present in Riverwood.
- Riverwood is a location. Riverwood is connected to Helgen.
- Helgen is a location. Helgen is connected to Whiterun.
- Whiterun is a location.
- Balgruuf is a character. Balgruuf is present in Whiterun.

No KG output: **Title:** The quest related to the above information is as follows:

Description: Find out what is going on in helgen

Tasks:

- Go to helgen.

Text2KG output: **Title:** A new order

Description: Help balgruuf the greater with his new order

Tasks:

- Arrive at whiterun.
- Find balgruuf the greater

Actual output: **Title:** Before the storm

Description: Inform balgruuf about the dragon in helgen

Tasks:

- Arrive at whiterun.
- Find balgruuf

Game: **Baldur's Gate 1**

Description: **Degrodel:** "First of all, my name is Degrodel. I have been on the lookout for skilled adventurers. My task is very simple.

Some years ago, an evil band of mercenaries were all petrified while trying to break into the home of Ramazith. For those of you with a limited vocabulary, petrified means "turned to stone." Anyhow, one of those adventurers had in his possession an item of power: the Helm of Balduran. Ramazith didn't know this at the time and so, after turning them all to stone, he simply sold their bodies to an art collector living in the heart of Baldur's Gate.

What I would like you to do is find these statues, turn them back to flesh, and take the item. I will give you these six scrolls of stone to flesh. Use them on the statues when you find them.

The estate can be found just west of the fairgrounds in the center of town. Good luck, and don't entertain any thoughts about betraying me. It could be very hazardous to your health.

You will be paid 5,000 gold for this service. This is a kingly sum, so I won't barter with you about the amount."

Plot points:

- mercenaries were petrified while trying to break into the home of Ramazith: Ramazith sold the mercenaries to an art collector
- Degrodel gives the player six scrolls of stone to flesh to unpetrify the mercenaries
- one of the mercenaries has the Helm of Balduran

KG2Text:

- Degrodel is a character. Degrodel is present in Baldur's Gate.
- Baldur's Gate is a location.
- the art collector's estate, west of the fairgrounds, in the center of Baldur's Gate is a location.
- Ramazith is a character. Ramazith is present in the art collector's estate, west of the fairgrounds, in the center of Baldur's Gate.
- Helm of Balduran is a object. Helm of Balduran is held by Degrodel. Helm of Balduran is present in the art collector's estate, west of the fairgrounds, in the center of Baldur's Gate.

No KG output: **Title:** The petrified mercenaries

Description: Unpetrify the petrified mercenaries

Tasks:

- Find ramazith's home.
- Give ramazith six scrolls of stone to flesh

Text2KG output: **Title:** The art collector's estate

Description: Bring the helm of balduran to ramazith

Tasks:

- Find the art collector’s estate, west of the fairgrounds, in the center of baldur’s gate.
- Kill the six mercenaries who were petrified while trying to break into ramazith’s home

Actual output: **Title:** Searching for balduran’s helmet

Description: Bring the helm of balduran to degrodel

Tasks:

- Proceed to the art collector’s estate, west of the fairgrounds, in the center of baldur’s gate.
- Unpetrify the mercenaries

Game: **Baldur's Gate 1**

Description: **Farthing:** "My name's Farthing. I'm only five. In a few years my papa will let me hunt. Are you allowed to hunt?

I've lost my dolly out in the woods. The gatekeeper won't let me go and get it. Can you get it for me?"

Plot points:

- the gatekeeper won't let Farthing go and get her doll

KG2Text:

- Farthing is a character. Farthing is present in the village.
- the village is a location.
- the woods is a location.
- Farthing's doll is a object. Farthing's doll is held by Farthing. Farthing's doll is present in the woods.

No KG output: **Title:** Farthing's doll

Description: Get farthing's doll

Tasks:

- Go to the cave.
- Find farthing's doll

Text2KG output: **Title:** Farthing's doll

Description: Bring farthing's doll back to farthing

Tasks:

- Find farthing's doll.

Actual output: **Title:** Farthing's dolly

Description: Return farthing's doll to her

Tasks:

- Travel to the woods.
- Find farthing's doll

Game: **Baldur’s Gate 2**

Description: **Korgan Bloodaxe:** "I’m scouring for a band of desperate adventurers to aid me in a gallant task. Come and hear Korgan Bloodaxe’s tale!

A fortnight past, me fellows and I were in the midst of obtaining an ancient text fer our patron when a skirmish visited our midst. Vile words, alas, became a lake of bloodshed. So it goes the sacred grimoire ne’er made it to our benefactor’s hand. And now he awaits its arrival with a zeal reserved for a grog-blossom in an alekeg. The scuttlebutt is the pay is handsome and worthy of note — sacks of loot and odd magics. The volume sits not far from ’ere, and the bibliomaniac, he paces the floor in the meantime. ’Tis called the Book of Kaza.

The Book of Kaza is in some hobnail’s tomb over in the lower crypts of the Graveyard District. It’s nae abandoned, but rumors and half-truths only scare the young and infirm. Keeps the curious away, me guess.

Well... it’ll take far more than a few shambling bags of skin and stitches to deter Korgan Bloodaxe from a king’s ransom! A foolhardy jaunt into a hive of undead? How could ye resist?"

Plot points:

- Korgan Bloodaxe and his fellows were on a mission to obtain the Book of Kaza for a book collector
- Korgan Bloodaxe and his fellows had a heated argument: only Korgan Bloodaxe remained alive afterwards

KG2Text:

- Korgan Bloodaxe is a character.
- a tomb over in the lower crypts of the Graveyard District is a location.
- Book of Kaza is a object. Book of Kaza is held by Korgan Bloodaxe. Book of Kaza is present in a tomb over in the lower crypts of the Graveyard District.

No KG output: **Title:** The book of kaza

Description: Bring korgan bloodaxe’s head to the book collector

Tasks:

- Travel to solitude.
- Find korgan bloodaxe

Text2KG output: **Title:** The book of kaza

Description: Get the book of kaza

Tasks:

- Reach a tomb over in the lower crypts of the graveyard district.

Actual output: **Title:** Help korgan recover the book of kaza

Description: Bring the the book of kaza to the book collector

Tasks:

- Head to a tomb over in the lower crypts of the graveyard district.
- Find the book of kaza with korgan bloodaxe

Game: **The Elder Scrolls: Skyrim**

Description: **Elder Othreloth:** "The Ancestral Tombs are where we inter our dead. They are sacred places, meant to honor their passing. Sadly, we've been unable to use the Ancestral Tomb beneath the Temple as of late, and have resorted to other locations to scatter the remains.

Those foul vermin... the abominations we call ash spawn have risen from the ashes of our own ancestors. They're befouling the memories of our forefathers by defiling their remains.

Could you cleanse the tomb for us, wanderer? That would be most kind. Here, this key should open the door to the tomb."

Plot points:

- Elder Othreloth gives the player a key that opens the door to the Ancestral Tomb

KG2Text:

- Elder Othreloth is a character. Elder Othreloth is present in Temple. Elder Othreloth is present in an Ancestral Tomb beneath the Temple.
- Temple is a location. Temple is connected to Ancestral Tombs.
- Ancestral Tombs is a location.
- an Ancestral Tomb beneath the Temple is a location.

No KG output: **Title:** Ancestral tomb

Description: Retrieve the ancestral blade

Tasks:

- Arrive at the ancestral tomb.
- Open the door with elder othreloth's key

Text2KG output: **Title:** Elder othreloth's secret

Description: Find elder othreloth's secret

Tasks:

- Travel to the temple.
- Find elder othreloth's secret

Actual output: **Title:** Clean sweep

Description: Cleanse the ancestral tomb

Tasks:

- Make your way to an ancestral tomb beneath the temple.
- Kill the ash spawn that defile the ancestral tomb

Game: **Torchlight II**

Description: **Fade:** "I am Fade - why have you summoned me?

Ah, you want to enter Cacklespit's hidden realm. Yes, I know the way - but I do not provide service without compensation.

I have very particular... appetites. The remains of the living are infused with a lingering residue of deeds past, which sustains me. Bring me the skulls of four warriors of renown, and I will take you where you wish to go. This battlefield is littered with the bones of great heroes. Your search should not be difficult."

Plot points:

- the player has summoned Fade: the player wants to enter Cacklespit's Realm

KG2Text:

- Fade is a character. Fade is present in Sundered Battlefield.
- Sundered Battlefield is a location. Sundered Battlefield is connected to Cacklespit's Realm.
- Cacklespit's Realm is a location.
- skulls of famous warriors is a object. skulls of famous warriors is held by Fade.

No KG output: **Title:** Fade

Description: Enter cacklespit's realm

Tasks:

- Travel to cacklespit's realm.

Text2KG output: **Title:** Skulls of famous warriors

Description: Find the skulls of famous warriors

Tasks:

- Travel to sundered battlefield.
- Find the skulls of famous warriors

Actual output: **Title:** Cacklespit's realm

Description: Bring 4 skulls of famous warriors to fade

Tasks:

- Find 4 skulls of famous warriors

Game: **The Elder Scrolls: Oblivion**

Description: **Martin:** "I've figured out another item needed for the ritual to open the portal to Camoran's Paradise. The second item is the counterpart to the first: the blood of a Divine.

This was a terrible puzzle to me. Unlike the Daedra, the gods have no artifacts, and do not physically manifest themselves in our world.

How then to obtain the blood of a god? But Jauffre solved it. The blood of Tiber Septim himself, who became one of the Divines. This is a secret remembered only by the Blades, passed down from one Grandmaster to the next.

Jauffre should tell it to you himself."

Plot points:

- Martin needs the blood of a Divine for the ritual to open the portal to Camoran's Paradise in addition to the first item, the blood of a Daedra

KG2Text:

- Martin is a character.
- Jauffre is a character.
- Tiber Septim is a character.

No KG output: **Title:** Divine blood

Description: Bring martin's ritual to a close

Tasks:

- Arrive at camoran's paradise.
- Obtain the blood of a divine

Text2KG output: **Title:** A ritual of blood

Description: Find a vial of the blood of a divine, a vial of the blood of a daedra, and a vial of the blood of a saint

Tasks:

- Proceed to the temple of the nine

Actual output: **Title:** Blood of the divines

Description: Recover the blood of a divine for martin

Tasks:

- Talk to jauffre about obtaining the blood of tiber septim

Game: **Baldur's Gate 2**

Description: **Flydian:** "I am Flydian, a messenger sent by the High Merchant of Trademeet to find someone to help our town.

Trademeet is under attack by forest animals. The High Merchant thinks druids may be responsible, but we haven't the manpower to find them and stop it.

The High Merchant will have more information for you."

Plot points:

- Flydian was sent by the High Merchant
- druids are suspected of the animal attacks

KG2Text:

- Flydian is a character. Flydian is present in Trademeet.
- Trademeet is a location.
- High Merchant is a character. High Merchant is present in Trademeet.

No KG output: **Title:** Finding flydian

Description: Find flydian

Tasks:

- Arrive at flydian's camp.

Text2KG output: **Title:** The high merchant's request

Description: Find out who is responsible for the animal attacks

Tasks:

- Find out who is responsible for the animal attacks.

Actual output: **Title:** Animal trouble in trademeet

Description: Investigate why forest animals are attacking trademeet

Tasks:

- Travel to trademeet.
- Go talk with the high merchant

Game: **Baldur's Gate 1**

Description: **Firebead Elvenhair:** "I, Firebead Elvenhair, have a small errand for you.

I left an identify scroll with Tethtoril, in the inner grounds. He should be done examining it by now, so if you could fetch it for me, I'd be grateful. I have a great use for those types of scrolls.

Allow me to cast a little spell on you as a reward. It will protect you from any evil you might meet tonight."

Plot points:

- Firebead Elvenhair left his identify scroll to Tethtoril

KG2Text:

- Firebead Elvenhair is a character.
- the inner grounds is a location.
- Tethtoril is a character. Tethtoril is present in the inner grounds.
- identify scroll is a object. identify scroll is held by Firebead Elvenhair. identify scroll is present in the inner grounds.

No KG output: **Title:** Firebead elvenhair

Description: Find firebead elvenhair

Tasks:

- Arrive at the goblin caves.
- Find firebead elvenhair

Text2KG output: **Title:** The identities of the elvenhair

Description: Bring the identify scroll to firebead elvenhair

Tasks:

- Find the inner grounds.

Actual output: **Title:** Firebead's scroll

Description: Bring firebead elvenhair's identify scroll to him

Tasks:

- Proceed to the inner grounds.
- Get firebead elvenhair's identify scroll from tethtoril