BookWorm: A Dataset for Character Description and Analysis

Argyrios Papoudakis **Mirella Lapata** **Frank Keller**
Institute of Language, Cognition and Computation
School of Informatics, University of Edinburgh
10 Crichton Street, Edinburgh EH8 9AB
a.papoudakis@sms.ed.ac.uk, {mlap, keller}@inf.ed.ac.uk

Abstract

Characters are at the heart of every story, driving the plot and engaging readers. In this study, we explore the understanding of characters in full-length books, which contain complex narratives and numerous interacting characters. We define two tasks: *character description*, which generates a brief factual profile, and *character analysis*, which offers an in-depth interpretation, including character development, personality, and social context. We introduce the BookWorm dataset, pairing books from the Gutenberg Project with human-written descriptions and analyses. Using this dataset, we evaluate state-of-the-art long-context models in zero-shot and fine-tuning settings, utilizing both retrieval-based and hierarchical processing for book-length inputs. Our findings show that retrieval-based approaches outperform hierarchical ones in both tasks. Additionally, fine-tuned models using coreference-based retrieval produce the most factual descriptions, as measured by fact- and entailment-based metrics. We hope our dataset, experiments, and analysis will inspire further research in character-based narrative understanding.

1 Introduction

Stories play a key role in shaping our understanding of the world, serving as a medium to share experiences, communicate ideas, teach, and entertain. The two main building blocks of every story are the plot and characters (Phelan, 1989). Characters are particularly important, as they form the primary means through which readers engage and relate to the story.

Understanding characters is also necessary from a computational perspective, if models are to summarize, analyse, or generate stories effectively. Over the past decade, the field of natural language processing has developed computational methods to understand narratives from a character-centric perspective. Previous work has focused on detecting characters (Chen and Choi, 2016), understanding latent personas (Bamman et al., 2013), their emotions (Kim and Klinger, 2019), and their relationships (Chaturvedi et al., 2017). Another line of work has attempted to describe characters with a set of attributes (Zhang et al., 2019) or personality types (Sang et al., 2022). Most prior research studies characters in short stories or adopts relatively simplistic analysis methods (e.g., summaries) when it comes to long narratives.

In this paper, we focus on analyzing characters in long-form stories, a relatively understudied area that presents unique challenges not found in short stories. Firstly, long stories typically contain a large number of characters with complex relationships and interactions which have a key role in the plot. Secondly, characters in long stories are dynamic (Chaturvedi et al., 2017): they develop throughout the story and their personalities, motivations, and relationships change as the plot evolves. Finally, long narratives exceed the input length that many current transformer-based architectures (Vaswani et al., 2017) can process, making the problem technically challenging.

We work towards addressing these challenges and study characters from a text-generation perspective, focusing on two tasks: (1) *character description* produces a general profile of a character (e.g., their actions, relationships, attributes) and (2) *character analysis* produces an in-depth interpretation of a character's personality and behaviour (e.g., how the character's personality develops, their motives, or the social context). The character description task has been introduced with the release of the LiSCU dataset (Brahman et al., 2021), which contains literary book summaries paired with human-written character descriptions. However, using summaries to describe characters significantly simplifies and restricts the task. Summaries contain limited information about the overall story, usually

Findings of the Association for Computational Linguistics: EMNLP 2024, pages 4471–4500
November 12-16, 2024 ©2024 Association for Computational Linguistics

***Book*:** Bleak House by Charles Dickens, ***Character*:** Esther Summerson

***Description*:** The narrator and protagonist. Esther, an orphan, becomes the housekeeper at Bleak House when she, Ada, and Richard are taken in by Mr. Jarndyce. Everyone loves Esther, who is selfless and nurturing, and she becomes the confidante of several young women. Although she eventually does find her mother, circumstances prevent them from developing a relationship. At first a hesitant, insecure narrator, Esther's confidence in her storytelling grows, and she controls the narrative skillfully.

***Analysis*:** Esther Summerson, the narrator and protagonist of Bleak House, [..] she proves to be a confident narrator who never misses the opportunity to relate others' compliments of her.[..] As her narrative gains breadth and depth, her confidence as a narrator grows. She deliberately withholds information or delays including it to give her story coherence and dramatic effect. And even though she is for the most part a reliable narrator (a narrator we can trust to accurately tell the story), she is less reliable when relaying information about her romantic life. Esther nurtures everyone around her, and her first instinct is to be motherly, perhaps because she has never had a caring mother figure of her own. [..] Ironically, Esther, for all her caring and tenderness, is the unwitting cause of great unhappiness. [..] Because of Esther's illegitimate birth, Lady Dedlock was forever estranged from her sister, Miss Barbary, and was forced to carry a painful secret. Because other unhappinesses, [..] we could argue that Esther is indirectly responsible for these as well.

Figure 1: Examples of character description and analysis. Both refer to the transformation of Esther Summerson from a hesitant to a confident narrator. However, the analysis provides more detail focusing on her skill as a narrator (red). The description includes Esther's attributes and behaviour, referring to her as a selfless and nurturing figure, while the analysis provides an interpretation of this trait based on her background (green). The character description briefly touches on Esther's background, while the analysis demonstrates how, ironically and indirectly, she causes pain to others (grey), adding a moral and psychological dimension.

only the salient events, and important details are omitted. At the same time, a book summary cannot be used to describe every character of a narrative, but only those important enough to figure in the summary.

For these reasons, our work focuses exclusively on describing characters attested in full-length books. In addition, we propose character analysis as a new task, which complements and extends character description in that it requires a more in-depth understanding of characters. It goes beyond just describing surface-level traits, critically analyzing the character's depth, complexity, and evolution within the narrative context. Character analyses also typically explore the social, political, or historical context relevant to understanding a character and their behaviour. We show an example of these two tasks in Figure 1. Additional examples can be found in Appendix A.

While previous work has made a significant effort to understand characters individually, treating them as isolated entities is a simplification. Characters have their own arcs in a story, but they are also interconnected – their actions, motivations, and relationships are all intertwined within the narrative (Weiland, 2016). Based on these observations, we introduce *Joint Character Description*, a variation of the character description task, where the model has to generate a description for *every* character sequentially. Our analysis shows that although current state-of-the-art language models can benefit from knowing all characters in a story, they struggle with joint character understanding.

Our contributions in this work are as follows:

- We propose BOOKWORM, a new dataset which enables fine-grained character comprehension for long-form texts and supports the tasks of character description (in isolation and jointly) and analysis.
- We establish baseline performance by training various state-of-the-art long context models, combined with different approaches to retrieving character information from long texts.
- Our experiments show that retrieval-based models lead to better performance on both tasks, despite hierarchical processing (Chang et al., 2024) being the de facto approach for book summarization in the literature.
- We expose limitations in the understanding capabilities of current models whose performance degrades when they attempt to reason about characters jointly.

2 Related Work

Narrative Structure Existing work has studied narratives and their plot structure, focusing primarily on summarization. Several datasets have been developed for narrative summarization. Examples include *TRIPOD* (Papalampidi et al., 2019), which contains movie scripts annotated with salient scenes or turning points. *NarraSum* (Zhao et al., 2022) has summaries of movies and TV series, while *BookSum* (Kryściński et al., 2021) is a collection of literary artefacts (e.g., novels, plays) paired with summaries. Summarization is related to our character description and analysis tasks, but there are significant differences (Brahman et al., 2021). A summary captures the entire plot of a story and includes *all* main characters, while a character description focuses on a *single* character, their properties and actions, incorporating plot elements only when they help describe the character.

Character Understanding Some prior research has studied narratives from a character-centric perspective, focusing on a variety of tasks: the identification of character personality (Sang et al., 2022; Bamman et al., 2013), prediction of character emotions (Brahman and Chaturvedi, 2020) and relationships (Chaturvedi et al., 2017), character detection (Chen and Choi, 2016), grounding (Liu and Keller, 2023), and the generation of character descriptions (Brahman et al., 2021). Another line of research has used characters as a means to generate new stories (Liu et al., 2020) or summaries (Zhang et al., 2019) of existing narratives. However, most existing research focuses on characters of short stories and studies only specific aspects (e.g., relationships, emotions). In this paper, we focus on characters of full-length books (with an average length of approximately 100k tokens), and study them more holistically, from the perspective of understanding their attributes, actions, and behavior (description task) *and* how are these interpreted in the context of the narrative (analysis task).

Character Description Previous work (Zhang et al., 2019) has found that character descriptions occur commonly in human-written story summaries, thus advocating the identification of a set of character attributes as a useful intermediate step for automatic summarization. Chen and Gimpel (2022) introduced *TVStoryGen*, a dataset aiming to generate TV episode recaps based on summaries and character descriptions. Carlsson et al. (2021) released *Gandalf*, a dataset containing descriptions paired with multiple character names, with the task of choosing the correct one. Brahman et al. (2021) created *LiSCU*, a dataset which contains book summaries and human-written character descriptions. A part of this dataset includes full books, however, it has not been publicly released and the work focuses on describing characters from summaries, not books. The current paper contributes to this literature by introducing BookWorm, a new dataset for understanding characters based on the full-length books, without assuming that summaries are available. We also introduce the new task of character analysis, which aims to generate a more detailed account of a character's personality, motives, development, and social context. We compare our dataset with LiSCU in Section 3.2.

Long-context Models Several papers have focused on alleviating the memory requirements of transformers, which increase quadratically with the input length. Sparse attention approaches like BigBird (Zaheer et al., 2021), Longformer (Beltagy et al., 2020), and Reformer (Kitaev et al., 2020) combine local windowed attention with global attention on a subset of tokens, enabling modelling of much longer sequences. Transformer-XL (Dai et al., 2019) introduces segment-level recurrence as another technique for capturing longer-range dependencies.

Another line of research has sought to overcome the limitations of input length by employing retrieval-augmented generation; Xu et al. (2023) show that retrieval can outperform long context transformers even when using shorter input. Other work (Wu et al., 2021; Chang et al., 2024) processes long documents hierarchically by segmenting the input into shorter chunks and generating intermediate responses, which are then aggregated into a final summary. We propose several retrieval-augmented models for our tasks, exploring different content extraction strategies (e.g., based on characters or a retrieval engine like BM25), and show they are superior to hierarchical generation.

3 The BookWorm Dataset

3.1 Data Collection

Following previous work (Kryściński et al., 2021; Brahman et al., 2021), we collect books from the Gutenberg Project[1], which contains classic books, including novels, plays, and works of poetry. To obtain character descriptions and analyses, we scrape five different websites, namely *Sparknotes*, *Litcharts*, *Gradesaver*, *Cliffnotes*, and *Shmoop*.[2] These websites contain complete studies of literary books, mainly for educational purposes. For our work, we use *Litcharts*, *Sparknotes*, *Gradesaver* and *Cliffsnotes* as sources for character descriptions and *Sparknotes*, *Shmoop* and *Cliffsnotes* as sources for character analyses. For websites that are used in both tasks, there is a clear distinction between the

[1] https://www.gutenberg.org/

[2] https://www.sparknotes.com/lit/, https://www.litcharts.com/, https://www.gradesaver.com/, https://www.cliffsnotes.com/, https://www.shmoop.com/

Dataset	Books	Samples	Avg. Characters	Avg. Input Length		Avg. Output Length	
				words	sentences	words	sentences
BookSum	187	405	—	108,477.13	5,195.34	1,151.86	54.84
LiSCU-summary	—	9,499	5.56	1,022.32	48.82	184.57	8.56
LiSCU-book	204	2,052	—	—	—	—	—
BookWorm (description)	324	5,869	9.74	97,685.82	4,481.16	88.79	3.97
BookWorm (analysis)	133	1,328	5.69	95,758.79	4,541.39	602.65	25.71

Table 1: Satistics for BookWorm and comparison with related datasets (LiSCU and BookSum). We show the total number of books, sample counts, and average length of input and output in terms of words and sentences.

different texts, for instance, Sparknotes contains a character list with descriptions and then more detailed and longer analyses entitled "in-depth analysis". For websites that are used only for description (e.g., Gradesaver) or analysis (e.g., Shmoop), we proceed to this selection after manually inspecting the corresponding text, their content, level of detail and length. To pair the books with their corresponding character descriptions and analyses, we match titles (after removing punctuation and lowercasing) and then manually verify that the authors are the same. We exclude books which belong to the genre "philosophy" as they do not have characters in the traditional sense. Additionally, we filter out character descriptions that are less than 30 words and character analyses that are less than 200 words to avoid short samples that do not fit the purposes of the tasks; usually, these correspond to minor characters in a story.

We split the data (for both tasks) into train/validation/test (80/10/10) partitions based on the book titles to avoid data leakage. All books in our dataset are in the public domain and do not have any copyright restrictions. We are not allowed to redistribute data from literature websites and thus use the Web Archive[3] to save scraped URLs, preserving the snapshot used in our experiments. We release our corpus to encourage future work on our tasks [4] (see examples in Figure 1 and in Appendix A).

3.2 Data Analysis

We present various statistics on BookWorm in Table 1 and compare it with the related LiSCU dataset (Brahman et al., 2021). For completeness, we also report statistics for BookSum[5] (Kryściński et al., 2021), a book summarization dataset.

Firstly, we note that the average book length for the description and analysis tasks is approximately 95k words, which is challenging even for current state-of-the-art transformer-based models. Additionally, we observe significant differences in task requirements; the average length of a description is 88 words, whereas the average analysis is 602 words. We have 324 and 133 unique books paired with 5,869 character descriptions and 1,328 character analyses, respectively. The LiSCU-summary partition has 9,499 samples, but contains only the summary of the story and not the full book; obtaining whole books is significantly harder than just collecting book summaries. Although the LiSCU-book partition is not publicly available, we report numbers from the corresponding paper (Brahman et al., 2021) in Table 1.

Following Kryściński et al. (2021), we show the literary genres represented in BookWorm in Figure 2. We observe similar trends across tasks: the books are mostly novels and plays, with some short stories, novellas and poetry collections; other genres (children's books, biographies and historical books) are sparsely represented.

4 Modeling Experiments

We conduct a series of experiments to benchmark the performance of current models and analyze their abilities across different dimensions. Initially, we explore the limits of simple extractive heuristics. Next, we evaluate an instruction-tuned model in a zero-shot setting and contrast its performance against fine-tuned models. Additionally, across our experiments, we evaluate different retrieval strategies that are generic or rely on domain-specific information and compare them with the hierarchical approach, which uses the full story.

4.1 Extractive Heuristics

We adopt the Lead-k baseline (Narayan et al., 2018), which traditionally extracts the first k sentences from a source document. We adjust this

[3] http://www.web.archive.org/

[4] https://github.com/apapoudakis/BookWorm

[5] We show statistics for the book-level partition.

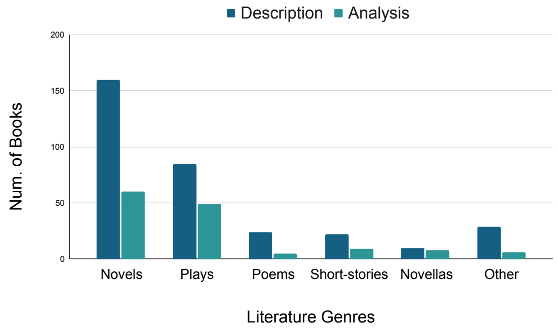

Figure 2: Distribution of genres in BOOKWORM for character description and analysis tasks.

baseline experiment to our task, extracting the first k sentences in which the character of interest is mentioned. We use the coreference model of the BookNLP library[6] to identify character mentions. Similarly, we define a Random baseline by randomly extracting k sentences in which the character is mentioned. To define an upper bound of the extractive experiments, we develop an Extractive Oracle baseline, selecting the k sentences with the highest average rouge score against the gold-standard (Narayan et al., 2018). For the description task, we set k equal to four, while for the analysis task, we set k to 25, based on the average number of sentences for each task in BOOKWORM.

4.2 Zero-shot Abstractive Models

All zero-shot experiments use Llama-3-8B-Instruct (Dubey et al., 2024) as a backbone model, with an input context of 8,192 tokens.[7] As a simple baseline, we feed the model with the lead 8,192 tokens, truncating the rest of the input.

We further develop retrieval-augmented models, following an extract-and-generate approach. In one variant, we use the statistical BM25 method (Robertson et al., 1995) to extract relevant context. Specifically, we use the character's name as a query and select the 80 paragraphs with the highest score. In another variant, we use a coreference model from the BookNLP library to identify character mentions and extract paragraphs in which the target character is mentioned, following prior work (Brahman et al., 2021; Maddela et al., 2022). We then concatenate these paragraphs and feed them into the language model.

We compare retrieval-augmented models to a hierarchical approach in which all the book information is processed. Following previous work (Wu et al., 2021; Chang et al., 2024), we split the book into chunks of 8k tokens and generate a description for each chunk. We then concatenate these intermediate descriptions and feed them to the language model, which merges them into a final description. We experimented with adding more steps to the hierarchical approach, but we did not observe an improvement, and thus only report the results of *single-step* hierarchical processing in this paper. We show the prompts used for our zero-shot models in Appendix B.

4.3 Fine-tuned Abstractive Models

We experiment with two architectures: the encoder-decoder LongT5-base model (Guo et al., 2022) with 16,384 tokens input context and the decoder-only Llama-3-8B-Instruct model (Dubey et al., 2024) with 8,192 tokens context length.

Analogously to our zero-shot models, we compare the fine-tuned models to a simple baseline, which truncates the input story at the maximum length the model can process. In addition, we evaluate the two retrieval strategies mentioned earlier, namely using BM25 or the coreference model from the BookNLP library. We fully fine-tune LongT5, while for Llama-3, we do parameter efficient fine-tuning using LoRA (Hu et al., 2021). We report the hyperparameters and additional training details in Appendix B.

4.4 Generation Settings

We report experiments in two settings. The first setting is common in previous work (Brahman et al., 2021) and aims to generate a description or analysis for a character in isolation. In addition, we explore an alternative formulation where we collectively describe or analyse all the characters in a story. We call this setting *joint character description*. We also explore a variant where the model has to describe every character separately, but all the characters from the story are given as input in the prompt, so as to ensure parity of context for the two alternative task formulations.

For the joint description setting, we employ the hierarchical approach in a zero-shot fashion as described in Section 4.2. We use Llama-3-70B-Instruct as the base model because we find that smaller models fail to describe the characters jointly and output descriptions for each. This is particularly problematic for books with many char-

[6] https://github.com/booknlp/booknlp

[7] We opted for 8k context as the model has been pre-trained with maximum sequences of this size.

acters. In this case, we adjust and prompt the model to describe five characters at a time instead of all characters together. If the model still struggles to follow the required template, then we describe the characters individually. We report the prompts used for these experiments in Appendix B.

4.5 Evaluation Metrics

Automated evaluation metrics are crucial for our task and for related book-length applications where human evaluation is extremely labor-intensive, costly, and difficult to design (Krishna et al., 2023). As there is no single agreed-upon metric for automatically measuring character understanding, we evaluate output quality along different dimensions and report several complementary metrics.

We use Rouge F1 (Lin, 2004) against the reference descriptions as a way of assessing the informativeness of descriptions or analyses. We report Rouge-1 (unigram overlap), Rouge-2 (bigram overlap), and Rouge-L (longest common subsequence between the model output and the gold-standard description). We also report entity mention recall following prior work (Bertsch et al., 2023), which counts the percentage of named entities (e.g., person names, locations) present in the reference that are covered by the model output. Additionally, we use BERTScore (Zhang et al., 2020), which calculates token similarity using contextual embeddings instead of string matching.

As token-matching evaluation does not always correlate well with the quality of the generated text (Fabbri et al., 2021), we also use QA-based evaluation, following existing literature (Deutsch et al., 2021; Fabbri et al., 2022). Specifically, we create a set of question-answer pairs based on the reference descriptions and then use the model output to answer these questions. We expect factual descriptions to correctly answer a higher percentage of questions. We first prompt GPT-3.5, asking it to generate question-answer pairs based on gold-standard descriptions. Since question-answering models are typically trained on data different from the narrative domain, such as Wikipedia passages, we fine-tune a RoBERTa-large encoder (Liu et al., 2019) using QA pairs from our dataset. We discard low-quality questions through round-trip filtering (Alberti et al., 2019), i.e., we check whether the generated questions can indeed be answered using the reference description. We employ exact match and F1 (Rajpurkar et al., 2016) to evaluate all QA models. We present examples of the question-answering evaluation in Appendix D.

As all the above metrics are reference-based, we also use an entailment-based metric, which predicts whether the input story entails the model output. Specifically, following Narayan et al. (2022) and Laban et al. (2021), for each generated sentence, we calculate its maximum entailment score against the paragraphs of the input story. If a paragraph is longer than 512 tokens, we split it into shorter paragraphs. We also transform the entailment probability into 0 or 1 using a 0.5 threshold. Then, we calculate the average entailment score across the model output. As an entailment model for our experiments we use T5-XXL (Raffel et al., 2020) fine-tuned on the Adversarial NLI dataset (Nie et al., 2020).

Previous research has also used LLM-as-a-judge pipelines to assess the quality of generated text (Mahon and Lapata, 2024; Min et al., 2023; Song et al., 2024; Zheng et al., 2023). In this paper, we adopt the PRISMA metric (Mahon and Lapata, 2024) using a large language model to evaluate the factuality of the generated outputs. Specifically, we calculate PRISMA-precision by extracting facts from the generated output and then using the gold-standard to judge whether these facts are supported or not. Similarly, we calculate PRISMA-recall by extracting facts from the gold-standard and then using the model output to assess these facts. PRISMA-F1 is then derived from these precision and recall values. We used GPT-4o-mini to extract facts and judge their factuality.

Additionally, we evaluate factuality across different character dimensions, by classifying the extracted facts into six distinct categories: Role (the part the character plays in the story), Relationship (connections the character has with others, e.g., friendships or family ties), Personality (the character's behavior, traits or attributes), Event (actions and decisions the character is involved in), Mental State (the character's state of mind, e.g., beliefs, intentions, and emotions), and Other Fact (any fact that does no belong to the above categories). We chose this categorization based on prior work (Brahman et al., 2021) and after having manually inspected examples of extracted facts. We again used GPT-4o-mini to classify facts into the above categories. To assess the reliability of the model's classification, we conducted a human annotation process where the authors of this paper classified 200 facts extracted from character descriptions and analyses. We found a strong agreement among the annotators with a Fleiss' Kappa of 74.64. We also found that GPT-4o-mini correlates highly with the majority of the human annotators, achieving a Cohen's Kappa of 62.47. Additional details for the fact-based evaluation are in Appendix B.

4.6 Implementation Details

All experiments were run on a single Nvidia A100 or H100 GPU, except for the joint description experiments, which used two H100s. We used pretrained models from HuggingFace and trained our models for four epochs selecting the checkpoint with the lowest validation loss. During inference, we used sample decoding with a temperature of 0.4. We report additional implementation details in Appendix B.

5 Results

There is no lead bias in book-length character understanding. Our experimental results are summarized in Table 2. Lead-k performs poorly, even though it is a strong baseline in standard summarization tasks (Nallapati et al., 2016; Narayan et al., 2018). It achieves substantially lower scores in terms of Rouge compared to zero-shot and fine-tuned models. Random selection performs similarly, achieving marginally worse scores than the Lead baseline in both tasks.

The extractive oracle heuristic achieves the highest Rouge scores across all experiments in both tasks. There is a bigger performance gap when it comes to the analysis task, where oracle experiment is ostensibly better, especially in Rouge-1 and Rouge-2, compared to zero-shot and fine-tuned models. This result is expected as the oracle model uses gold-standard texts to extract sentences. When considering Entity Mention recall, we observe that the Oracle model is worse at the description task than zero-shot and fine-tuned models. Interestingly, in the analysis task, while the Oracle model scores lower than zero-shot models in Entity Mention recall, it surpasses the fine-tuned models in this metric. This result demonstrates that there is still space for improvement in the way that our experiments retrieve context and use salient entities.

BERTScore results for the Oracle model are comparable to the Lead and Random heuristics, and lower compared to abstractive models. This is an expected outcome, as the Oracle fails to capture the semantic information of the reference descriptions, even if it matches the gold-standard tokens.

Retrieval-augmented models perform best in both character description and analysis. In our zero-shot experiments, we observe that the retrieval-based methods consistently improve performance in both tasks. Specifically, the coreference approach outperforms BM25 in the description task while BM25 performs better in the analysis. The hierarchical approach improves results compared to the Lead baseline in the description task but does not match the performance of retrieval-based methods. In the analysis task, the performance is slightly worse than in the Lead experiment. We hypothesize that this occurs because retrieving relevant information is more crucial than processing the entire story for tasks like character description and analysis, which resemble query-based summarization more than generic summarization.

For our fine-tuned models, we observe trends similar to the zero-shot ones. Specifically, both the BM25 and coreference-based retrieval lead to better descriptions and analyses, with the exception of LongT5 in the analysis task, where the differences are only marginal. Llama-3 consistently outperforms LongT5 across both tasks. While fine-tuning leads to consistent improvements in the description task, this is not the case for the analysis task where fine-tuning is either comparable or inferior to the zero-shot setting. We hypothesize that there are two reasons for this, the training samples are fewer in the analysis task and the level of data contamination is higher (see Appendix C).

Fine-tuned Llama with coreference-based retrieval is the most faithful. We report QA-based and NLI-based evaluation results in Table 3. We focus on fine-tuned models as these performed better in most cases than zero-shot ones and extractive baselines, according to reference-based metrics (see Table 2). QA-based metrics reward Llama most when enhanced with coreference-based retrieval in both tasks. In general, performance improves when relevant context is retrieved in both the LongT5 and Llama models. Llama consistently outperforms LongT5, and coreference-based retrieval yields better results than extraction using BM25.

The NLI metric has a clear preference for models fine-tuned on coreference-based input. In particular for Llama, we observe a large jump in entailment accuracy over BM25. Retrieving relevant context helps achieve higher entailment scores for both tasks. The only exception is the entailment accu-

		Description					Analysis				
	Model	R-1	R-2	R-L	EntMent	BS	R-1	R-2	R-L	EntMent	BS
Heuristic	Lead	20.64	2.08	12.23	24.08	49.33	25.20	2.22	10.96	12.89	45.87
	Random	19.67	1.65	11.58	22.62	49.34	25.00	2.22	10.91	11.68	45.82
	Oracle	34.38	8.97	21.46	21.46	50.75	42.16	14.03	17.86	21.13	48.74
Zero-shot	Llama-3	27.89	5.00	17.24	28.15	55.21	32.48	6.18	16.34	20.70	54.16
	+ BM25	29.69	5.95	18.28	31.60	57.29	33.59	**6.68**	**16.75**	**24.22**	54.56
	+ coref	29.86	5.98	18.55	32.27	**57.90**	33.36	6.60	16.60	22.80	**54.70**
	+ hier	28.46	5.67	17.72	29.44	56.74	31.47	6.09	15.69	23.16	53.79
Fine-tuning	LongT5	28.62	5.53	17.91	26.73	54.91	33.03	6.27	15.38	11.86	50.01
	+ BM25	29.98	5.84	18.61	28.69	56.27	32.19	6.02	15.49	12.80	48.98
	+ coref	29.19	5.84	17.82	31.15	55.43	32.38	6.13	15.18	10.27	49.56
	Llama-3	27.90	5.62	18.74	29.59	55.34	33.59	6.32	15.49	16.60	52.86
	+ BM25	**29.78**	6.42	19.67	34.80	56.93	**34.10**	6.53	15.62	18.09	52.87
	+ coref	**30.36**	**6.75**	**19.84**	**35.78**	57.63	33.93	6.55	15.67	18.28	53.10

Table 2: Results on character description and analysis tasks on our BOOKWORM dataset. We use Rouge, entity mention recall (EntMent) and BERTScore (BS). Best model per metric is boldfaced (excluding the oracle).

	Description			Analysis		
Model	EM	F1	NLI	EM	F1	NLI
LongT5	8.69	12.20	7.94	6.17	7.78	7.25
+ BM25	9.62	12.83	10.28	6.75	8.33	2.84
+ coref	9.53	13.78	16.10	7.14	9.25	13.33
Llama-3	8.16	12.04	16.87	5.58	8.39	14.15
+ BM25	10.41	15.28	22.70	6.94	11.19	13.06
+ coref	**13.29**	**17.36**	**40.21**	**8.49**	**11.27**	**50.85**
Reference	–	–	65.27	–	–	61.43

Table 3: Question answering and entailment-based evaluation for fine-tuned models. We report exact match (EM), F1, and natural language inference metric (NLI). Best model per metric is boldfaced.

racy of LongT5 combined with BM25 on the analysis task, where performance decreases compared to LongT5 on its own. The coreference resolution approach is consistently better than BM25.

Facts related to events and relationships are hard to get right. We report the fact-based evaluation in Table 4. We observe that retrieval-augmented models demonstrate higher overall factuality, leading to improvements across nearly all character dimensions for both tasks. An exception is the LongT5 model for the character analysis task, where the lead baseline outperforms BM25 and coreference-based models. The coreference-based model surpasses BM25 in description and analysis, while the Llama model consistently outperforms LongT5. Across both tasks, facts related to events and character relationships are the least factual. In contrast, facts concerning a character's role and personality achieve the highest scores. Mental state and other facts perform similarly, but they fall below those related to personality and role. Our results demonstrate that models struggle with the more dynamic aspects of characters, such as events and relationships, while handling more static dimensions like role and personality more effectively. Examples of the fact-based evaluation are in Appendix D.

It is easier to talk about one character than about many. Table 5 presents results in the two generation settings: joint and separate character description.[8] For this comparison, we employ the hierarchical method in a zero-shot setting with Llama-3-70B-instruct (see Section 4.4), as we observed that smaller models could not follow instructions for the joint task.

The model generally struggles with the joint task, performing consistently worse across metrics compared to describing each character individually. As we can see in Table 5, the model benefits from having a list of the characters in the story. We observe performance gains across metrics when character names are included in the input. We believe the joint description task is too difficult for the model which is now required to understand the story from beginning to end instead of being able to focus on a single character. Aside from understanding being

[8] We do not perform joint experiments for the analysis task, as this would be extremely challenging.

	Description							Analysis						
Model	Role	Relat.	Person.	Event	Mental S.	Other	Overall	Role	Relat.	Person.	Event	Mental S.	Other	Overall
LongT5	19.90	16.68	29.71	10.49	23.25	22.23	19.98	26.49	17.83	28.63	13.20	23.52	19.73	22.54
+ BM25	28.51	21.03	32.00	13.56	25.31	24.72	22.34	26.15	17.70	24.90	12.19	18.69	25.06	20.60
+ coref	33.41	26.43	32.33	14.53	25.72	28.14	25.46	24.29	18.32	25.15	11.08	18.88	20.69	19.72
Llama-3	38.77	20.99	40.61	27.02	32.70	37.25	34.89	48.53	34.17	42.11	**29.57**	35.59	37.60	36.56
+ BM25	51.88	38.99	51.84	34.04	39.80	45.41	42.80	50.71	33.34	44.11	29.29	**37.35**	37.82	37.20
+ coref	**53.66**	**41.31**	**53.50**	**34.54**	**43.39**	**47.91**	**52.68**	**52.12**	**36.65**	**46.11**	29.48	36.85	**39.28**	**37.95**

Table 4: PRISMA evaluation results on description and analysis tasks for the fine-tuned models. We report the overall PRISMA-F1 score along with the F1 score of each subcategory (Role, Personality, Relationship, Event, Mental State and Other Fact). Best model per metric is boldfaced.

Model	R-L	EntMent	QA-F1
Separate	17.82	29.92	14.37
+ character names	**18.36**	**31.71**	**14.63**
Joint	16.62	23.39	12.78

Table 5: Character description results in separate and joint generation settings. We report Rouge-L, entity mention recall (EntMent) and question answering F1. We use the hierarchical approach with zero-shot Llama-3-70B-Instruct. Best model per metric is boldfaced.

harder, generation is also more challenging, as the output is quite long in this setting. Even Llama-3-70B struggles to describe *all* the characters. Examples of generated outputs are in Appendix D.

6 Discussion

Our experiments underline the importance of retrieving relevant context; we found that even simple methods such as statistical retrieval with BM25 or coreference-based retrieval lead to consistent improvements in all our experiments. Notably, while the hierarchical approach is considered state-of-the-art for book summarization, our experiments revealed it performs worse than retrieval in both description and the analysis tasks. The difference between retrieval-based and hierarchical approaches is significant even for character analysis. One might conjecture that processing the whole book would be beneficial, however, this is not corroborated by our results.

Until now, characters have been studied separately in the literature, which is a significant simplification. Our experiments with joint understanding of characters show that a separate description model can benefit from knowing all the different characters in a story if we list them in the initial prompt. However, our experiments also demonstrate that models struggle to understand characters jointly, having to "read" a book multiple times to be able to describe each character separately.

7 Conclusions

In this work, we created BookWorm, a new dataset which contains books from the Gutenberg project and human-written character descriptions and character analyses from literature websites. Character descriptions are short and factual, while character analyses are longer; they explore the motives, personality, and development of a character and often also comment on the social, historical, or political context. We established a set of baselines using simple extractive heuristics as well as retrieval-based and hierarchical long-context models, in both zero-shot and fine-tuning settings. Our experiments highlight the importance of retrieving relevant context, which leads to consistent improvements and outperforms hierarchical methods.

We hope our findings will inspire future research on character analysis, and text generation from long documents more generally. We plan to develop a better suited model for the joint character description task, by keeping track of characters and their relations as the narrative evolves. Evaluation is another avenue for future work. Entailment-based metrics are good indicators of model performance for retrieval-augmented approaches, but are computationally challenging for book-length inputs. Question-answering evaluation helps assess the factuality of the generated text but is constrained by the availability of references, and can be too punitive (in cases where model predictions have no lexical overlap with the reference). We used a LLM as a judge to perform a fact-based evaluation and gain a deeper understanding of the factuality of the different character dimensions. However, these results are again solely based on reference texts. In the future, there is a need to explore evaluation metrics that consider the full input text and are at the same time efficient and scalable.

Limitations

Our dataset contains publicly available books discussed widely across multiple sources (reviews, critical essays, literary commentaries, etc). Even if models have not been trained on the description and analysis tasks, it is likely that they have been exposed to these literary texts or related information during pre-training. To mitigate the risk of data contamination, future work should consider using books that are not publicly available.

In this paper, we relied on automatic metrics such as Rouge, QA-based evaluation, entailment and fact-based scores, entity mention recall and BERTScore to assess the quality of generated descriptions and analyses; however, the majority of these metrics are reference-based and do not consider book-length input to evaluate different aspects of model output. Future work could focus on reference-free evaluation metrics and efficient methods to conduct human-based evaluation.

Moreover, current language models used in this paper do not provide any explanation about the generated text. Future research could focus more on attributable language models that generate text pointing to specific parts of the input. This would also mitigate the difficulty of conducting human evaluation, especially for tasks like character description or analysis, where many responses can be produced, but it is important to evaluate whether they are faithful.

Our work considers simple retrieval-based strategies such as BM25 and the use of an off-the-self coreference model. A natural next step would be to explicitly train a retriever model for the two tasks in the BOOKWORM dataset. Finally, we present experiments with only one type of hierarchical model in the joint character description setting. Follow-on work could study this setting in more depth.

Acknowledgements

This work was supported in part by the UKRI Centre for Doctoral Training in Natural Language Processing, funded by the UKRI (grant EP/S022481/1) and the University of Edinburgh, School of Informatics and School of Philosophy, Psychology & Language Sciences. Lapata gratefully acknowledges the support of the UK Engineering and Physical Sciences Research Council (grant EP/L016427/1).

References

Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. 2019. Synthetic QA Corpora Generation with Roundtrip Consistency. *arXiv preprint*. ArXiv:1906.05416 [cs].

David Bamman, Brendan O'Connor, and Noah A. Smith. 2013. Learning Latent Personas of Film Characters. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 352–361, Sofia, Bulgaria. Association for Computational Linguistics.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The Long-Document Transformer. *arXiv preprint*. ArXiv:2004.05150 [cs].

Amanda Bertsch, Uri Alon, Graham Neubig, and Matthew R. Gormley. 2023. Unlimiformer: Long-Range Transformers with Unlimited Length Input. *arXiv preprint*. ArXiv:2305.01625 [cs].

Faeze Brahman and Snigdha Chaturvedi. 2020. Modeling Protagonist Emotions for Emotion-Aware Storytelling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5277–5294, Online. Association for Computational Linguistics.

Faeze Brahman, Meng Huang, Oyvind Tafjord, Chao Zhao, Mrinmaya Sachan, and Snigdha Chaturvedi. 2021. "Let Your Characters Tell Their Story": A Dataset for Character-Centric Narrative Understanding. *arXiv preprint*. ArXiv:2109.05438 [cs].

Fredrik Carlsson, Magnus Sahlgren, Fredrik Olsson, and Amaru Cuba Gyllensten. 2021. GANDALF: a General Character Name Description Dataset for Long Fiction. In *Proceedings of the 3rd Workshop on Machine Reading for Question Answering*, pages 119–132, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yapei Chang, Kyle Lo, Tanya Goyal, and Mohit Iyyer. 2024. BooookScore: A systematic exploration of book-length summarization in the era of LLMs. *arXiv preprint*. ArXiv:2310.00785 [cs].

Snigdha Chaturvedi, Mohit Iyyer, and Hal Daume Iii. 2017. Unsupervised Learning of Evolving Relationships Between Literary Characters. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1). Number: 1.

Mingda Chen and Kevin Gimpel. 2022. TVStoryGen: A Dataset for Generating Stories with Character Descriptions. *arXiv preprint*. ArXiv:2109.08833 [cs].

Yu-Hsin Chen and Jinho D. Choi. 2016. Character Identification on Multiparty Conversation: Identifying Mentions of Characters in TV Shows. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 90–100, Los Angeles. Association for Computational Linguistics.