Using Longformer with Seq2Seq and Coreference Resolution to Generate Character Analyses from Full Texts

Ajay Bati (abati7@gatech.edu)
Nyimul Hoque (nhoque3@gatech.edu)
Rahul Komatineni (rkomatineni6@gatech.edu)
Georgia Institute of Technology CS 4650: Natural Language

Abstract

In this paper, we present a method of approach to create summaries of characters' stories in literary pieces, based on coreference resolution, training a Longformer model (with Seq2Seq), reinforcement learning, and character relationship graphs. We created an example dataset using coreference resolution including context with Harry Potter and the Philosopher's Stone. We extracted data from the LiSCU dataset, which is a dataset of literary pieces and their summaries paired with descriptions of characters that appear in them (presented by "Let Your Characters Tell Their Story": A Dataset for Character-Centric Narrative Understanding). We present our findings with the results from training the seq2seq hugging face model using the LongformerEncoderDecoder (LED) from Allen et. al. on the LISCU dataset and additionally evaluating on the Harry Potter coreference dataset. We also present our findings and results using reinforcement learning and generating character relationship graphs.

1 Introduction

The ability to automatically generate summaries of texts has been a long-standing goal that is actively researched in the field of natural language processing. One of the key challenges in this task of summarization is identifying the salient characters and their relationships to the plot, and more specifically their relationships to each other. This is abstractive summarization.

This paper presents our approach to analyzing characters' arcs in literary pieces using natural language processing techniques. Specifically, we utilize coreference resolution, training a LongformerEncoderDecoder (LED) model (with Seq2Seq), and reinforcement learning to generate character-centric summaries. We also create character relationship graphs and present those findings.

We trained the LongformerEncoderDecoder (LED) model to generate character descriptions based on the summary of the entire book. We access this data from LiSCU (Brahman et al., 2021). We hope that the model can learn to generate descriptions of a character given some larger context (summary, actions, etc.). Then, we create mock summaries of a given character for each chapter using coreference resolution. We feed these summaries into the trained LED to understand the state of a character for every chapter.

Additionally, we incorporate reinforcement learning to fine-tune the generated summaries and improve their coherence and readability. We hypothesize that much of a given summary for a query character will contain miscellaneous, unneeded information. To remove this, we devise a binary classification reinforcement learning approach similar to (Narayan et al., 2018). At the end, we use this classifier to remove unnecessary sentences from each of our mock summaries in hopes of increasing the accuracy of the generated character description.

To evaluate the effectiveness of our approach, we also created a dataset using coreference resolution with Harry Potter and the Philosopher's Stone.

This project is for our team's final research project in the Natural Language (C.S. 4650) class at Georgia Institute of Technology. It is an implementation based on the culmination of what we learned in class and our own research. The rest of this paper is organized as follows: we discuss related Engines in section 2, followed by an explanation of the Dataset in section 3. This is followed by a detailed explanation of our methodology in the Coreference Resolution, Longformer Model, and Reinforcement Learning sections (sections 4, 5, and 6 respectively). We present our experimental Results in Section 7, and conclude with sections 8 through 10 which are titled Limitations, Ethics Statement, and Acknowledgement respectively.

2 Engines

To generate our dataset, we used an M1 Mac and shell scripts to extract data from the LiSCU dataset and perform coreference resolution with each Harry Potter novel. We used the spaCy library to perform this coreference resolution. We also utilized Python to preprocess the data and prepare it for training the model.

To train our models, along with the reinforcement learning and character graph generation, we used a Google Colab Pro subscription, which offers access to Nvidia Tesla K80, T4, P4, and P100 GPUs. These GPUs provide a significant performance boost compared to using only a CPU, allowing us to train our models much faster. For our models, we used the Hugging Face transformers library to implement our Seq2Seq models and the reinforcement learning algorithm. The data and model are further discussed in the next section, and limitations and challenges that we ran into with respect to these engines and libraries are discussed in section 6.

3 Dataset

The data we used is called the LiSCU dataset. It is described as "a dataset of literary pieces and their summaries paired with descriptions of characters that appear in them (Brahman et al., 2021)." More specifically, we extracted .json files, and the relevant columns were labeled "Character Name," "Summary," and "Description," where the summary is the same per literary piece, but the description pertains to the character for that data entry. The goal is to feed these summaries into a language model with a character name and train it to produce an abstractive summary of the query character.

We also created a custom dataset of what we called "makeshift" summaries that essentially index through a literary piece's text, looks for a certain character's name, and grabs the surrounding few sentences (i.e. starting from the sentence that is 75 characters behind the character name to the sentence that is 75 characters ahead the character name.). To capture more references of a given character, we first run coreference resolution on the text and replace all mentions of that character.

4 Coreference Resolution

In order to find all the mentions of a character in each chapter, we used coereference resolution. We utilized AllenNLP's SpanBERT model (Joshi et al., 2020) that has shown state-of-the-art (SOTA) performance for coreference resolution.

For our project, it was essential that we figure out every time a certain character is referenced or mentioned, and coreference resolution allows for us to replace pronouns and nicknames with character names. This way, we are able to create mock summaries for a given character by parsing the text, finding mentions of this characters, and appending 50 characters behind and following this mention for context.

5 LongFormer Model and Results

There are several encoder decoder models that have empirically worked really well for the summarization task. However, they are not able to perform equally as well for longer query texts. The LiSCU dataset presents us with summaries with average token length of 1024, but there are around 20% that are more than 3000 tokens in length. Models such as BART and GPT2 are not suited for more than 1024 tokens. The authors in LiSCU (Brahman et al., 2021) use these models along with a Longformer encoder layer. However, we try to experiment with entire encoder-decoder version of the LongFormer.

Therefore, we choose the Longformer encoder-decoder architecture as our character description generator. With its local windowed attention mechanism along with global attention, it can capture much depedencies acorss much larger text (Beltagy et al., 2020). This adaptation increases runtime efficiency as well. Global attention is the same as traditional attention and will span across all tokens while local attention is localized around a window of tokens.

Our input to the model is the character name along with the entire book summary: **name** *sep* **summary**. We make sure to apply global attention on both the character name and the sep token (this is suggested from huggingface for summarization tasks). We train this model on two sets hyperparameters. The first, baseline model was trained with 2048 maximum token length for summaries and 256 maximum token length for gen-

Model\Dataset	Validation Dataset
LEDv1	0.2745/0.09221/0.1390
LEDv2	0.2601/0.0857/0.1292

Model\Dataset	Harry Potter Summary
LEDv1	0.07804/0.03461/0.04545
LEDv2	0.0654/0.0215/0.0368

Table 1: Fmeasure for dataset and each model in the format rouge1/rouge2/rougeL

erated descriptions. The second model was larger and was trained with 4096 maximum token length for summaries and 512 maximum token length for generated descriptions. The output average fmeasures of the Rouge1, Rouge2, and RougeL are displayed in the table.

The recall of these results are what drag the model's performance down. Sometimes, the character descriptions have less than 20 tokens while other times they have over 500. Much of the information present in these descriptions do not generally overlap exactly with the summaries provided. Therefore, model's generated character description will often contain other hallucinated content. On the other hand, we found the precision to be relatively high compared to recall for this similar reason. Information overlap exists, but much of this information is also hallucinated.

6 Reinforcement Learning and Results

The summaries that we use both in the LiSCU dataset as well as our mock summaries contain various sentences that have no relevance to a given character. If we feed these summaries into the trained LED, it may confused the model. Therefore, we hope to resolve this confusion by removing sentences that are less relevant.

For any given character and summary, we do not have labeled data telling us about each sentence's relevance. Therefore, we hope to train a reinforcement learning agent using REINFORCE (Williams, 1992). We separate a summary into separate sentences and encode each sentence using SentenceBERT (Reimers and Gurevych, 2019). We feed these embeddings into a TransformerEncoder along with an embedding of the character name. At the end, we place a classification head that outputs probabilities of whether or not to include that sentence in a summary. We sample

Model\Dataset	LISCU Train Dataset
RL Baseline	0.135/.061/0.103

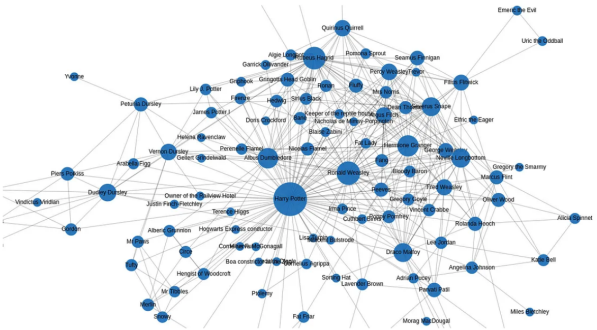
Table 2: Fmeasure for dataset and each model in the format rouge1/rouge2/rougeL

these distributions and get all the sentences which were sampled as 1. This forms our mock character description. We compare it to the gold character description and record this as our reward. To update our parameters, we use a similar approach as seen in (Narayan et al., 2018) in section 4.

We can see the results of the character descriptions generated from this model in Table 2. This is just an experimental section and the results seem promising as the model picks sentences that often pertain to the query character.

7 Character Graph

To further understand the role of each character, we developed a character graph for Harry Potter and the Philospher's Stone. Each character is given a node where the size of each is proportional to their importance. Importance in this context was measured as the number of times a character was mentioned in the text. We used both coreference resolution on the original text and used aliases for characters from the official Harry Potter fandom page to be able to count each time a character appears, or how important they are. We also drew edges if a character ever explicitly interacted with another character (Sims and Bamman, 2020). We did not consider the number of times an interaction occurred or about implicit interactions. The graph is shown below. As expected, Harry Potter has the biggest node and is in the near center of the graph as he has the most connections to characters in the book and therefore has the most edges. We also added a full size version in the appendix.



8 Limitations

We ran into a few challenges and limitations when working on this project. The main one being scalability. We worked with the LED model because of the Longformer's ability to take in up to 16k tokens using a sliding window technique. However, the first Harry Potter novel (one of the books we worked very closely with) alone is 76,944 tokens long. As a result, we went chapter by chapter to complete our methodology.

Even with access to powerful GPUs, we still faced computational and time constraints. We had to reduce our batch size and input sequence length and output token length more than we would have preferred, to fit the available GPU memory and storage constraints. We also had to carefully manage the training process to ensure that we made efficient use of the available resources, for example, by using early stopping to prevent overfitting and reduce training time.

Despite these challenges, we were able to train our models and generate character-centric summaries for the literary pieces in our dataset. Had we been given more time, we would have done more training, hyperparameter tuning. Coreference resolution would also be improved, and the reinforcement learning aspect of our project would have been more fleshed out.

9 Ethics Statement

As with any technology, natural language processing can have ethical implications that should be carefully considered. In our approach to create summaries of characters' stories in literary pieces, we recognize the importance of respecting privacy: all the data we worked with is publicly available data and does not pertain to any individuals' information. We also acknowledge the potential for misuse or unintended consequences of this application of NLP, and we strive to be mindful of the impact of our work. As we continue to develop and refine our methods, we will remain vigilant in our ethical considerations and seek to contribute to constructive dialogue and reflection on the responsible use of NLP and artificial intelligence more broadly.

10 Acknowledgements

We would like to acknowledge our Professor Wei Xu and the very heplful team of TAs for their guidance in helping our understanding on NLP models and in answering specific questions we had about the process in building our dataset and model. The LiSCU data and Longformer model were integral to helping us complete this project.

References

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv:2004.05150 [cs]*.

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:2109.05438 [cs]*.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *arXiv:1907.10529 [cs]*.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Ranking sentences for extractive summarization with reinforcement learning. *arXiv:1802.08636 [cs]*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Matthew Sims and David Bamman. 2020. Measuring information propagation in literary social networks. *arXiv:2004.13980 [cs]*.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Reinforcement learning*, pages 5–32.

11 Appendix

We have a full scale picture for readability.