Creating Fiction Through De-summarization

Anonymous EMNLP submission

Abstract

While story generation is on the rise, it seems that most of the effort is put into producing very short stories. We aim to change that by introducing an approach to gradually de-summarize a short prompt, such as a title, into a longer text. We present 3 models all of which beat the baseline language model in human evaluation. In addition to the models, we also present a new automatic metric based on the NLI task. This metric aims to assess the consistency of a text.

1 Introduction

In the last couple of years, there have been some promising breakthroughs in the field of open-ended story generation (Fan et al., 2018; Clark et al., 2018; Ammanabrolu et al., 2019). However, these works often focus on producing very short stories, such as those in the ROCstories dataset (Mostafazadeh et al., 2017).

In our work, we are exploring a 'reverse summarization' approach in order to generate stories. In the summarization task, the input is the article/document and the output is a brief summary of that article/document. What we decided to do is to de-summarize, meaning we flip the inputs and outputs. The input is the summary (in our case, title of a movie/play) and output is the article (in our case, a story).

While we are not yet ready to generate full-length plays, we have had success with producing longer stories. Since it is not possible to read every machine-generated output and evaluate it manually, we also present a new metric based on the NLI task which focuses on text consistency.

2 Dataset

Our dataset contained three types of data:

- short stories (480 words on average)
- long stories (900 words on average)
- full-length scripts

Type of Data	Examples
Short summaries	∼40K
Long summaries	∼1.5K
Scripts	∼1.6K

Table 1: Amount of data

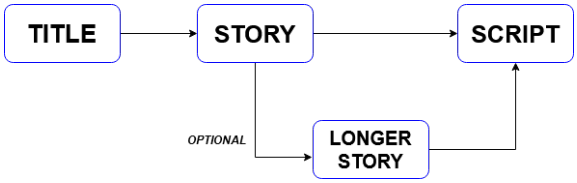

Figure 1: Hierarchical generation approach

We used two existing datasets: Wiki Movie Plots dataset (Robischon, 2018) which contains short summaries of movies from Wikipedia and Movie Plot Synopses with Tags dataset (Kar et al., 2018) which contains short summaries of movies from both Wikipedia and IMDb. The rest of the data we used has been compiled during the work on the THEaiTRE project (Rosa et al., 2020) and contains short and long theatre play summaries from Wikipedia, Sparknotes and various fan-sourced websites. This portion of data also contains the full-length scripts of movies and episodes of TV shows taken from fan-sourced script collections such as the IMSDb[1] or Forever Dreaming Transcripts.[2]

During the preprocessing phase, the data was cleared of duplicities, corresponding summaries were merged with the scripts, and the data was separated into a train and test split with a ratio 9:1. The absolute amounts of examples can be seen in Table 1.

[1] https://imsdb.com/

[2] https://transcripts.foreverdreaming.org/

3 Approach

Language models are not yet capable of producing a full-length play by themselves (Rosa et al., 2020) and often produce texts which are uninteresting and heavily repetitive (See et al., 2019). Our goal was to develop a pipeline which would allow us to progressively generate longer texts without serious declines in coherence and consistency.

With this goal in mind, we present our hierarchical approach. As illustrated in Figure 1, we aim to break the generation down into a couple of steps:

1. Produce a short story when given a story title.
2. Optionally elaborate on the short story and turn it into a longer one.
3. Separate the stories into sentences or other blocks
4. Use the blocks to generate short scripts
5. Concatenate the short scripts into a single script

These steps are inspired by the data we have. We have alignments between all the proposed transitions, however the amount of aligned data containing scripts was not sufficient to make the model work. So far, we can present models involved in steps 1 and 2. Further dataset improvements will have to be made in order to successfully produce models which can generate scripts.

4 Models

We fine-tune the following three models:

- GPT-2 (Radford et al., 2019) was developed by OpenAI for text generation. GPT-2 is a large transformer based model with 1.5 billion parameters. It generates synthetic text in response to the model being prompted with an arbitary input.
- PEGASUS (Zhang et al., 2019) was developed by researchers at Google AI for abstractive summarization. It achieved state of the art performance on 12 summarization tasks.
- DistilBART (Shleifer and Rush, 2020) was developed by researchers at Facebook AI for practical use of large scare pre-trained transformers. They apply 'shrink and fine-tune' method to original BART (Lewis et al., 2019) and PEGASUS.

5 Evaluation

Since this work is one of the first in its kind, there are no established evaluation metrics and procedures yet. For this reason, we use some well-established metrics (such as perplexity or ROUGE score), perform manual evaluation, and propose our own automatic metric based on the NLI task. We have used 500 examples generated from the test set in our automatic evaluation process.

5.1 Supervised Metrics

We use two automatic evaluation metrics; ROUGE score (Lin, 2004) and BERTScore (Zhang* et al., 2020). We call them supervised, because both of them need a reference text. ROUGE measures n-gram precision, recall, and f-measure between candidate and reference text. It is presently mostly used for evaluating summarization systems. BERTScore measures the cosine similarity in contextual embeddings between candidate and reference text. It was proposed in 2020 as an improvement to previous automatic evaluation metrics for text generation.

Results In Table 2 we can observe that GPT-2 performed considerably well compared to PEGASUS and DistilBART. This is a bit surprising since PEGASUS and DistilBART are Seq2Seq models so we hypothesized that they will be able to encode the prompts better.

However, the problem with ROUGE score is that it is an n-gram matching based evaluation. So we also use BERTScore which captures the position and context of the word by computing cosine similarity using contextual word embeddings. In Table 3 we can see that the GPT-2 is again performing better than DistilBART and PEGASUS but only by a few decimals. This indicates that all three of them were able to generate semantically correct sentences

5.2 Unsupervised Metrics

Since we would like to be able to evaluate open-ended story generation with potentially no reference text, it is desirable to suggest metrics which do not require it. All of the average values of the metrics and characteristics listed below can be found in Table 4. By no means do we consider the following list of metrics to be exhaustive.

Sentence length We have elected to observe sentence length, because texts containing sentences

Model	Precision	Recall	F-measure
GPT-2	26.65	30.83	22.12
PEGASUS	26.70	6.04	8.35
DistilBART	32.14	13.30	15.44

Table 2: ROUGE uni-gram scores

Model	Precision	Recall	F-measure
GPT-2	81.84	81.80	81.80
PEGASUS	82.54	80.13	81.29
DistilBART	82.06	80.99	81.51

Table 3: BERTScore

that are too long tend to be more difficult to read. On the other hand, sentences are too short increase the monotonicity of a text, making it duller. In order to get a better idea about what values we should aim for, we also ran the evaluation on the test set which contains human-written stories. In Table 4 we can see that all models are very close to the average human sentence length with DistilBART being the closest.

Story length As we mentioned in the introduction, most of the previous work focuses on short stories that are around 5 sentences long. Since our goal is to generate longer texts, it makes sense to try to capture just how much longer they are. Of course, this can be influenced via a decoding parameter, but our models can also choose to end generating before reaching the maximum length. We report the length in sentences and words. Table 4 shows that GPT-2 is capable of writing the longest summaries and therefore is the closest to the golden lengths we wish to achieve.

Perplexity Perplexity is a standard metric for any natural language generation task. If it is too low, it can indicate that the text is repetitive or uninteresting. In the opposite case, when the perplexity is unusually high, it could potentially mean that the text does not adhere to a proper grammatical structure. Looking at Table 4, we can find it strange that stories generated by GPT-2 have a higher perplexity on average. This could mean that GPT-2 is producing unlikely and/or original stories. However, this number alone is not enough to determine whether that hypothesis is true. It is necessery to assess the behavior of the model using human evaluation.

5.3 NLI-based Consistency Metric

The natural language inference (NLI) task is to determine whether a given sentence is entailed in, neutral to, or in contradiction with a piece of text. This is useful for us, because we want to avoid inconsistencies – contradictions. At the same time, entailment is also not preferable, because it can indicate repetition. Essentially, we are interested in how neutral a given sentence is in relation to the preceding text. This neutrality is computed by using the roberta-large-mnli model by (Liu et al., 2019).

Instead of just approaching this task as a classification, we apply softmax over the logits given by the model in order to obtain a distribution of the categories. Then we take the probability of the 'neutral' category as the basis for our score. We are using this approach instead of counting the occurence of classified categories, because we found that on our data, the model almost never openly states that a sentence is a contradiction or an entailment – neutral is always prominent. However, the changes in distribution usually do reflect the requested result.

In order to make the metric resistant to reasonable changes in the length of evaluated text, we propose to measure the average neutrality per added sentence. The second sentence is compared with the first, the third with the first two, and so on. Once the chunk of preceding text became too long to fit into the model, we truncated it by removing entire sentences from the beginning.

5.3.1 Results of the NLI-Score

We can see the results of this evaluation in Table 5. According to the metric, each of our models performs on a similar level than the human-written stories. However, it is necessary to point out that in case of DistilBART and PEGASUS, we found that the models sometimes produced the same story many times – regardless of the title. These stories scored pretty high in our similarity metric and therefore are pushing the average values up.

It is also important to look back at Table 4 and realize that DistilBART and PEGASUS produced much shorter stories on average. Naturally, it is easier not to run into any contradictions or entailments when the story is very short.

5.3.2 Limitations of the metric

As we can see, even the human-written stories do not score as high as we had originally expected. This can be due to two reasons:

Error in NLI classification The model which classifies the sentences into the NLI categories is

Model	Length	Sentences	Words	Perplexity
Test set	21	482	25	2.41
GPT-2	19	377	20	3.15
PEGASUS	18	53	3	1.31
DistilBART	20	107	6	1.54

Table 4: The average values of automatic metrics

Model	NLI-Score	Std
Test set	0.73	0.16
GPT-2	0.68	0.1
PEGASUS	0.68	0.16
DistilBART	0.67	0.13

Table 5: Average NLI-Score and the standard deviation

Model	1st	2nd	3rd
Baseline	4	6	8
GPT-2	5	9	4
PEGASUS	8	5	5
DistilBART	7	4	7

Table 6: The relative ranking results

not a 100% accurate. It is quite possible that it makes mistakes, especially if a contradiction is very subtle.

Is neutrality what we want? We have described our reasons for wanting the generated sentences to be as neutral as possible to the previous ones. However, this might not be the optimal approach. Perhaps plot twists are meant to contain contradictions, or some level of entailment can improve the quality of the story.

5.3.3 Potential usage

In a human-in-the-loop scenario, this metric can be used for filtering out stories that are objectively not consistent. We would recommend setting the threshold somewhere between 0.3 - 0.5, based on the individual's preference.

5.4 Manual Evaluation

Since evaluating almost any natural language generation task is very subjective and cannot yet be captured by automatic metrics, it is necessary to perform manual evaluation. Apart from our fine-tuned models described in Section 4, the manual evaluation also featured a vanilla GPT-2-medium as the baseline. We carried out two manual evaluation procedures:

5.4.1 Relative ranking

The annotators were asked to look at 3 stories at a time. Their task was to order them from the best to the worst according to their own subjective judgement with no further instructions. In total, we had 6 annotators looking at 12 stories in 4 comparisons.

As can be seen in Table 6, every one of our models outperformed the baseline in this task. GPT-2 was most frequently the "middle option" and was considered to be the worst the least amount of times. According to our annotators, the best stories were written by Pegasus. An interesting phenomenon occured while evaluating DistilBART - there were 2 annotators who selected DistilBART as their top choice whenever it was present in the comparison.

5.4.2 Absolute scoring

The annotators were shown one story at a time and were asked to evaluate it in terms of:

- Coherence – Is the text coherent?
- Consistency – Are the characters consistent?
- Originality – Is the text original and/or interesting?
- Title Relevance – Is the title relevant to the story?
- Overall Impression – Did you enjoy reading this text?

Each of the attributes was rated on a 5-point scale with 1 being the worst, and 5 being the best.

In this evaluation task, GPT-2 proved to be the best all-around model. It only slightly underperforms the baseline in the title relevance ranking. We hypothesize that this is the case, because our models were trained on data which included more abstract relations between the title and the story. The baseline then uses the titles more literally than our models.

Model	Coherence	Consistency	Originality	Relevance	Overall Impression
Baseline	2.7	2.8	2.6	2.7	2.6
GPT-2	3	3.1	3.1	2.6	3.2
PEGASUS	2.8	2.8	3	2.1	2.8
DistilBART	1.9	2	3.2	2	2.9

Table 7: The average absolute scores

6 Conclusion

We have presented an approach to generate longer texts which are still conditioned on an input prompt – in our case the title. By gradually de-summarizing the title, we hope to eventually arrive at a computer generated full-length script.

We have trained and evaluated 3 separate models. Based on the evaluation, we consider our fine-tuned GPT-2 to be the most useful model for our continued attempts to generate a theatre play.

We have also presented a new metric for natural language generation based on the NLI task. We are also hoping to improve a possible human-in-the-loop scenario by using this metric as a filter.

As future work, we are hoping to devise a similar system for evaluating even longer stories. Moreover, we are hoping to bridge the gap between the stories we have now and the scripts they could become.

References

Prithviraj Ammanabrolu, Ethan Tien, Wesley Cheung, Zhaochen Luo, William Ma, Lara Martin, and Mark Riedl. 2019. Guided Neural Language Generation for Automated Storytelling. In *Proceedings of the Second Workshop on Storytelling*, pages 46–55, Florence, Italy. Association for Computational Linguistics.

Elizabeth Clark, Yangfeng Ji, and Noah A. Smith. 2018. Neural Text Generation in Stories Using Entity Representations as Context. In *Proceedings of NAACL-HLT*, pages 2250–2260, New Orleans, Louisiana. Association for Computational Linguistics.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical Neural Story Generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, New Orleans, LA, USA. ArXiv: 1805.04833.

Sudipta Kar, Suraj Maharjan, A. Pastor López-Monroy, and Thamar Solorio. 2018. MPST: A corpus of movie plot synopses with tags. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France. European Language Resources Association (ELRA).

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Nasrin Mostafazadeh, Michael Roth, Annie Louis, Nathanael Chambers, and James Allen. 2017. LSDSem 2017 shared task: The story cloze test. In *Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics*, pages 46–51, Valencia, Spain. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. Technical report, OpenAI.

Justin Robischon. 2018. Wikipedia movie plots.

Rudolf Rosa, Ondřej Dušek, Tom Kocmi, David Mareček, Tomáš Musil, Patrícia Schmidtová, Dominik Jurko, Ondřej Bojar, Daniel Hrbek, David Košťák, Martina Kinská, Josef Doležal, and Klára Vosecká. 2020. Theaitre: Artificial intelligence to write a theatre play. In *Proceedings of AI4Narratives — Workshop on Artificial Intelligence for Narratives*, pages 9–13, Aachen, Germany. RWTH Aachen University.

Abigail See, Aneesh Pappu, Rohun Saxena, Akhila Yerukola, and Christopher D. Manning. 2019. Do massively pretrained language models make better storytellers? In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 843–861, Hong Kong, China. Association for Computational Linguistics.

Sam Shleifer and Alexander M. Rush. 2020. Pre-trained summarization distillation. *CoRR*, abs/2010.13002.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2019. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

A Samples of Generated Stories

A.1 GPT-2

The Haunted House Mystery The film is set in the late 19th century, in the small town of Fairfield, New Hampshire, where the story is told. The story begins with a young woman named Susan, who is haunted by the ghost of her deceased mother, who died in a car accident. Susan and her family are forced to move into a dilapidated house on the outskirts of town. Susan's mother died in an automobile accident, leaving her with no memory of what happened to her mother. Susan begins to remember the events of her mother's death, including the car accident that killed her mother, and the death of her stepfather, who was also killed in the accident. She begins to suspect that the ghost may be responsible for the deaths of her father and stepmother.

A.2 DistilBART

An empty bag and a dead Eve in the Tunnel The Tunnel Troll has been searching for a new identity since it was pulled over by the racist, homophobic, and sexist Sheriff Dollard. He discovers the truth behind the infamous "drag princess" incident in the tunnel. While searching for the new identity of the drag queen, Dollard is ridiculed for her actions by her colleagues, who believe she was a drag queen.

A.3 Pegasus

Three and Two is Five During the reign of Queen Elizabeth I, England is concerned by the impending arrival of the Spanish Armada. In 1588, relations between Spain and England are at breaking point. With the support of Queen Victoria, English privateers such as Sir Francis Drake regularly capture Spanish merchantmen bringing gold from the New World. Elizabeth's chief advisers are the Lord Treasurer, Lord Burleigh, and her longtime admirer, Robert Dudley, Earl of Leicester. Burleigh's 18-year-old granddaughter Cynthia is one of Elizabeth' three widows, and the ageing queen is plagued by guilt over the girl's alleged involvement in prostitution.