Two-Stage Curriculum Fine-Tuning for Abstractive Summarization of Video-Game News and Reviews

Winter 2025

Lingqi Huang
Department of Statistics
University of Michigan -Ann Arbor
hlingqi@umich.edu

Abstract

Leveraging game-news articles from IGN, GameSpot, and Polygon, this project develops an abstractive summarization system. We gather data via web scraping and official APIs, then apply a two-stage curriculum fine-tuning strategy to several pretrained encoder–decoder models from Hugging Face-T5, Longformer Encoder-Decoder (LED), Pegasus, and BART. By experimenting with model sizes from 60 million to 770 million parameters and evaluating on ROUGE and BERTScore metrics, we demonstrate that Pegasus-large performs best, achieving a BERTScore F1 of 0.952 and a ROUGE-2 score of 0.7204. The source code can be found here :https://github.com/KingOliver666/Final-project.git

1 Introduction

The video-game market has grown into a vast and dynamic ecosystem, with U.S. consumers alone spending $58.7 billion in 2024 across hardware, software, and in-game content. Navigating this explosion of titles—across genres from open-world epics to competitive shooters—poses a challenge for both newcomers, who must sift through voluminous coverage to find games that match their tastes, and veteran players, who often look for concise comparisons of the latest releases.

Already, major console and franchise announcements are driving a surge in news volume. Nintendo is set to launch its next-generation hybrid, the Nintendo Switch 2, on June 5, 2025 (preorders begin April 24 at an MSRP of $449.99), Rockstar Games has scheduled Grand Theft Auto VI for release in Fall 2025 on PlayStation 5 and Xbox Series X|S, and CD Projekt RED has confirmed that The Witcher IV will not arrive until at least 2027. This flood of high-profile launches underscores the need for tools that can distill lengthy articles into their key insights.

To address this, we propose developing an abstractive summarization system tailored to video-game news and reviews. By leveraging transfer learning on pretrained encoder–decoder architectures—such as T5, LED, Pegasus, and BART—we aim to capture domain-specific terminology and writing styles. Our approach employs a two-stage curriculum fine-tuning regimen: in the first stage, models learn from shorter, simpler summaries to establish core summarization skills; in the second stage, they're exposed to longer, more complex articles to master nuanced, game-specific language. We fine-tune models of varying sizes (60 M–770 M parameters) on a curated dataset from IGN, GameSpot, and Polygon, evaluating performance with ROUGE and BERTScore to identify the most effective configurations.

2 Problem Definition

We seek to build an abstractive summarization system tailored to video-game news. Concretely, we define:

- $\mathcal{A} = \{a_1, \dots, a_N\}$ be a set of game-news articles(from IGN, GameSpot, Polygon).
- $\mathcal{S} = \{s_1, \dots, s_N\}$ be the corresponding human-written reference summaries(or machine produced summaries).
- $f_\theta : \mathcal{A} \to \hat{Y}$ be our encoder-decoder model parametrized by θ, which produces a generated summary $\hat{y}_i = f_\theta(a_i)$.

Our learning objective is to find:

$$\theta^* = \arg\max_\theta \frac{1}{N_{val}} \sum_{(a_i,s_i)\in\mathcal{D}_{val}} M(f_\theta(a_i), s_i)$$

where M is our evaluation metric like ROUGE or BERTScore, and $\mathcal{D}_{val}$ is the held-out validation set.

2.1 Evaluation Metrics

We focus on two metrics-ROUGE-N and BERTScore F1-to evaluate summary quality both at the surface (n-gram) and semantic(embedding) levels.

2.1.1 Rouge-N

Formally, ROUGE-N is an n-gram recall between a candidate summary and a set of reference summaries. ROUGE-N is computed as follows(Ganesan, 2018):

$$\text{ROUGE-N} = \frac{\sum\limits_{s \in \mathcal{S}} \sum\limits_{gram_n \in s} Count_{match}(gram_n)}{\sum\limits_{s \in \mathcal{S}} \sum\limits_{gram_n \in s} Count_{(}gram_n)}$$

where n stands for the length of the n-gram, $gram_n$ and $Count_{match}(gram_n)$ is the maximum number of n-grams co-occuring in a candidate summary and a set of reference summaries.
We emphasize ROUGE-N(N = 1 or 2) to capture the unigram and bigram overlap between generated and reference summaries.

2.1.2 BERTScore

We define(Zhang et al., 2020b):

- $C = \{c_1, \ldots, c_{|C|}\}$ be tokens of candidate $\hat{y}$.
- $R = \{r_1, \ldots, r_{|R|}\}$ be tokens of reference s.
- $E(t) \in \mathbb{R}^d$ be the embedding of token t from pretrained transformer.

We compute the cosine-similarity matrix:

$$S_{i,j} = \cos(E(c_i), E(r_j))$$

and then we compute the precision and recall as:

$$P_B = \frac{1}{|C|} \sum_{i=1}^{|C|} \max_j S_{i,j}, \quad R_B = \frac{1}{|R|} \sum_{j=1}^{|R|} \max_i S_{i,j}$$

then the BERTScore F1 is computed as:

$$F1_B = \frac{2P_B R_B}{P_B + R_B}$$

The BERTScore F1 captures soft semantic alignment between candidate and reference at the embedding level.

By optimizing θ to maximize these metrics on validation data, we aim to produce concise, faithful summaries of game-news articles that preserve both surface-level content(ROUGE-1 and ROUGE-2) and deeper semantic content(BERTScore)

3 Data

The dataset for this project consists of 1,000 recent game-news articles sourced from IGN, GameSpot, and Polygon via the NewsAPI. Because of the API's daily rate limit of 100 articles per source, we sampled uniformly across platforms until we reached our target size, ensuring coverage of Xbox, Switch, PC, and PS5 topics. During preprocessing, we removed any article whose text exceeded 2,000 words to keep input lengths tractable for our encoder–decoder models. In future work, we plan to expand the corpus—either by upgrading to a paid NewsAPI plan or by supplementing with additional web-scraped data—to further improve our summarizer's performance in the gaming domain.

One example game news article would look like below:

Nintendo Switch 2 includes an all-new GameShare feature. Previously, players could pass a second Joy-Con to a friend and play together locally, but now that feature has been extended to let you play locally with another Nintendo Switch 2 system with just one copy of a game. Even "games that require multiple screens" can be played if one person owns a compatible game. You can temporarily share a game with up to three other local systems at a time, including OG Nintendo Switch. GameShare also support online play with other Switch 2 systems, although it's only available with compatible games such as Captain Toad Treasure Tracker, Super Mario 3D World, Super Mario Odyssey, Big Brain Academy, and Club House Games 51 Worldwide Classics, which is getting a free update on Switch. Please note, however, Nintendo Switch 2 exclusive games can only be shared with Nintendo Switch 2 system.

A major challenge in our dataset is the absence of human-written "gold" summaries for game-news articles—recruiting annotators to produce them would be prohibitively time-consuming and expensive. To bootstrap a pseudo–ground-truth, we leveraged a pretrained BART-large model (406 M parameters) fine-tuned on the CNN/DailyMail corpus. We passed our entire pool of game-news articles through this model and generated synthetic reference summaries, constraining them to a maximum length of 350 words and a minimum of 120 words. These machine-generated summaries serve as stand-in labels for subsequent fine-tuning and evaluation of our domain-specific summarizers.

4 Related Work

4.1 Summarization tasks

Automatic text summarization has evolved from early extractive approaches to today's powerful neural, abstractive systems.

• **Extractive Summarization**: Early work framed summarization as a sentence-selection problem. (Luhn, 1958) pioneered frequency-based scoring of salient terms, while (Edmundson, 1969) introduced cue-phrase and positional features to rank sentences. These heuristic methods, though simple, laid the groundwork for selecting representative content from documents.

• **Neural Abstractive Summarization**: The advent of sequence-to-sequence (seq2seq) models with attention (Sutskever et al., 2014)(Bahdanau et al., 2016) enabled end-to-end generation of novel summaries. (Rush et al., 2015) applied this paradigm to the CNN/DailyMail dataset, demonstrating that encoder–decoder architectures could produce fluent abstracts. Subsequent advances—such as the pointer-generator network (See et al., 2017)—combined copying and generation to balance faithfulness and fluency.

4.2 Pretrained Transformer Models

Nowadays with the rise of large pretrained Transformer models, summarizations has shifted toward fine-tuning versatile, self-supervised architectures:

• BART (Lewis et al., 2019): A denoising autoencoder that corrupts text (e.g. token masking, sentence shuffling) and learns to reconstruct the original. Its encoder–decoder setup makes it naturally suited for summarization once fine-tuned on paired article–summary data.

• T5 (Raffel et al., 2020): Reformulates every NLP task—translation, classification, summarization—as "text-to-text" by prepending a task prefix (e.g. "summarize:"). T5's unified framework and extensive pretraining corpus enable strong zero- and few-shot performance.

• PEGASUS (Zhang et al., 2020a): Introduces a novel pretraining objective ("Gap Sentence Generation") that deletes whole sentences and trains the model to generate them, closely mirroring abstractive summarization. This task-tailored pretraining yields state-of-the-art results on many summarization benchmarks.

• LED (Beltagy et al., 2020): Extends the Longformer's attention mechanism to encoder–decoder settings, allowing efficient processing of long documents (up to several thousand tokens) by combining global and sliding-window attention patterns—critical for summarizing lengthy game-news articles.

4.3 Two-Stage Fine-Tuning

Curriculum learning theory posits that models train more effectively when tasks are organized from simple to complex. (Sotudeh et al., 2023) demonstrate this in abstractive summarization by re-weighting and re-ordering training instances from "easy" to "hard," yielding faster convergence and higher ROUGE/BERTScore on the CNN/DailyMail corpus.

we adopt a length-based, two-stage fine-tuning curriculum within a transfer-learning framework: first, we fine-tune pretrained encoder–decoder models (BART, T5, Pegasus, LED) on game-news articles to generate concise summaries of up to 250 tokens, enabling the model to learn core content selection; next, we continue training the Stage 1 checkpoint to produce more detailed summaries of up to 350 tokens, thereby gradually increasing output complexity. By applying this regimen across architectures ranging from 60 M to 770 M parameters and evaluating on a held-out validation set using ROUGE-2 and BERTScore, we systematically investigate trade-offs between model capacity, computational cost, and summarization quality. Our results show that this curriculum fine-tuning strategy not only stabilizes training but also consistently boosts performance—paralleling successes in domain-specific summarization of news, scientific articles, and legal documents.

5 Methodology

Our summarization piplines follows a two-stage fine-tuning regimen applied uniformly to each pretrained encoder-decoder model(BART, T5, Pegasus, LED). The staged approach yields more stable training, helps the model first master concise abstraction, and then gradually adapts to produce richer, longer summaries.

5.1 Data Preprocessing and Tokenization

Each document is tokenized and either truncated or padded to a fixed input length of $L_{in} = 512$

tokens. Fine-tuning then follows a two-stage curriculum: in Stage 1, summaries are constrained to a maximum length of $L_{sum}^{(1)} = 250$ tokens; in Stage 2, the summary ceiling is raised to $L_{sum}^{(2)} = 350$ tokensto enable richer, more detailed outputs. Throughout both stages, we invoke each model's tokenizer in "target" mode to separately encode the reference summaries, and we use a uniform batch size of B=8 for all training and evaluation.

5.2 Two-Stage Fine-Tuning

Our fine-tuning strategy employs a two-stage curriculum to adapt pretrained encoder–decoder models to the game-news domain. In Stage 1, we initialize the model with its pretrained weights and fine-tune it for $E^{(1)} = 20$ epochs, truncating or padding each input to $L_{\text{in}} = 512$ tokens and constraining generated summaries to a maximum of $L_{\text{sum}}^{(1)} = 250$ tokens. We optimize the cross-entropy loss under teacher-forcing with a uniform batch size of $B = 8$ and perform evaluation on the validation split at regular intervals. Upon completion, we carry the learned weights into Stage 2, where we relax the output constraint—allowing summaries up to $L_{\text{sum}}^{(2)} = 350$ tokens—and continue fine-tuning for $E^{(2)} = 20$ epochs under the same batching and optimization settings. This length-based curriculum enables the model to master concise abstractions before progressively handling more detailed summaries, resulting in more stable convergence and improved ROUGE-2 and BERTScore performance.

5.3 Evaluation Metrics

We then evaluate our fine-tuned model on the held-out validation set using ROUGE-1, ROUGE-2, ROUGE-L, and BERTScore F1. All metrics are computed after decoding tokens IDs into text, skipping special tokens. ROUGE captures surface-level overlap, while BERTScore assesses deeper semantic alignment.

6 Experiment Result

In this section, we evaluate the performance of our two-stage fine-tuned summarization models on a held-out validation set of game-news articles. We report quantitative results using our chosen metrics (ROUGE-1, ROUGE-2, and BERTScore F1). All training and evaluation were performed on a server(Great Lakes) equipped with one NVIDIA A100 (80 GB) GPUs and one NVIDIA Tesla V100 (16 GB) GPU, with 128 GB of RAM.

Table 1: Summarization performance on the validation set for stage 1.

Model	ROUGE-1	ROUGE-2	BERTScore F1
T5-small	0.177	0.116	0.844
BART-base	0.215	0.178	0.871
LED-base	0.229	0.191	0.873
Pegasus-large	**0.736**	**0.644**	**0.939**
T5-large	0.22	0.181	0.872

Table 2: Summarization performance on the validation set for stage 2.

Model	ROUGE-1	ROUGE-2	BERTScore F1
T5-small	0.185	0.132	0.857
BART-base	0.221	0.184	0.873
LED-base	0.227	0.186	0.873
Pegasus-large	**0.795**	**0.72**	**0.952**
T5-large	0.219	0.182	0.871

The results in Tables 1-2 reveal several key insights. First, among all architectures, Pegasus large outperforms the others in both stages, achieving a ROUGE-2 F1 of 0.644 in Stage 1 and rising to 0.720 in Stage 2 (with BERTScore F1 improving from 0.939 to 0.952). This dominance underscores the effectiveness of Pegasus's gapsentence pretraining for abstractive summarization in the gamenews domain.

Second, the two-stage curriculum yields consistent gains for some models but not all. BART-base improves modestly-ROUGE-2 from 0.178 → 0.184 (+3.4 %) and BERTScore from 0.871 → 0.873—and T5-small sees larger relative improvements (+13.8 % in ROUGE-2, +1.5 % in BERTScore). In contrast, LED-base and T5-large exhibit negligible or slightly negative changes across stages, suggesting that models with stronger initial summarization performance (Pegasus-large) or specialized pretraining objectives derive the greatest benefit from a length-based curriculum.

Third, model capacity alone does not guarantee superior results: T5-large (770 M parameters) lags behind LED-base (436 M) in both ROUGE-2 and BERTScore, highlighting the importance of pretraining task alignment. Pegasus's task-specific objective yields much higher gains than simply scaling model size.

Finally, the parallel improvements in ROUGE and BERTScore affirm that our two-stage regime not only increases surface n-gram overlap but also enhances deeper semantic fidelity. Future work could explore more granular curricula (e.g. intermediate summary lengths), dynamic difficulty schedules, or domain-adaptive pretraining to further uplift models—especially those that showed limited gains under the current two-stage setup.

7 Conclusion

In this work, we have presented a domain-specific abstractive summarization system tailored to video-game news and reviews. Leveraging a novel length-based, two-stage fine-tuning curriculum, we adapted four pretrained encoder–decoder architectures (BART, T5, Pegasus, and LED) to the gaming domain. By first training on concise summaries (≤ 250 tokens) and then on more detailed outputs (≤ 350 tokens), our approach stabilizes training and yields consistent gains in both ROUGE-2 and BERTScore across model sizes ranging from 60 M to 770 M parameters. Empirical results demonstrate that Pegasus-large, in particular, benefits most from this curriculum, achieving a ROUGE-2 F1 of 0.7204 and a BERTScore F1 of 0.952 on our held-out validation set.

Our findings highlight two key insights: (1) curriculum fine-tuning over summary length effectively guides models from basic abstraction to richer generation, and (2) pretraining objectives aligned with summarization (e.g. Pegasus's gap-sentence generation) can outperform mere scale increases. While our machine-generated pseudo-references enabled rapid development, future work should incorporate human annotations or semi-supervised refinement to further improve fidelity. We also plan to expand our corpus via paid APIs and web scraping, explore more granular curricula (e.g. intermediate summary lengths or difficulty-based sampling), and conduct user studies to assess real-world utility. Overall, this study demonstrates that carefully structured transfer learning can produce high-quality, efficient summaries in specialized content domains.

8 Limitations and Future Work

Despite the promising results, our study has several limitations. First, the size of our training corpus ($N < 1000$ articles) is relatively small for fine-tuning large Transformer models. This scarcity may lead to under-fitting and prevent the models from fully capturing domain-specific patterns in video-game journalism. In future work, we plan to expand the dataset by upgrading to a paid NewsAPI tier, scaling up web-scraping pipelines, and incorporating additional sources (e.g. GameInformer, Eurogamer). We will also explore semi-supervised approaches that leverage unlabeled articles to augment training.

Second, our pseudo-ground-truth summaries were generated by a medium-sized BART-large teacher (406 M parameters) due to inference time and memory constraints on our cluster. As a result, the quality of these labels may limit the student models—especially larger ones such as T5-large—from reaching their full potential. To address this, future work could employ model distillation from more powerful teachers (e.g. Pegasus-xlarge or GPT-style models) and investigate iterative self-training loops, where student outputs are gradually refined by successive teacher generations.

Third, we adopted only two fixed summary lengths ($L_{\text{sum}}^{(1)} = 250$ and $L_{\text{sum}}^{(2)} = 350$ tokens) due to limited GPU time. It remains unclear how intermediate or more extreme length constraints might affect model performance and the optimal curriculum schedule. In follow-up studies, we will conduct a finer-grained ablation over multiple target lengths (e.g. 150, 300, 450 tokens) and explore dynamic, instance-level curricula that adapt summary length based on article complexity or topic.

A further limitation lies in our exclusive reliance on automatic metrics (ROUGE-2 F1 and BERTScore F1) computed against machine-generated pseudo-references. Because these pseudo-summaries may contain errors, omissions, or stylistic biases, models that closely mimic them can achieve high automatic scores without producing genuinely high-quality abstractions. As a result, T5-large—which may generate more coherent or informative summaries—can be unfairly penalized if its outputs deviate from flawed pseudo-reference patterns, while Pegasus-large benefits from closer alignment with those labels. Automatic metrics alone therefore may not reflect human-perceived quality. In future work, we will perform comprehensive human evaluations—rating summaries on coherence, relevance, and fluency via crowd-sourced or expert annotations—to better calibrate automatic scores and ensure that model selection aligns with end-user preferences.

Looking ahead, several additional avenues can build on our work. Domain-adaptive pretraining on a large unlabeled corpus of game-related text could further align model representations with gaming terminology. Human evaluation and error analysis—e.g. assessing factual accuracy, topical coverage, and stylistic fidelity—will be crucial to validate metrics such as ROUGE and BERTScore. Finally, integrating our summarizer into real-world gaming platforms or news aggregators, and measuring its impact on user satisfaction and reading efficiency, will provide a more comprehensive assessment of its practical utility.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2016. Neural machine translation by jointly learning to align and translate. https://arxiv.org/abs/1409.0473.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. https://arxiv.org/abs/2004.05150.

H. P. Edmundson. 1969. New methods in automatic extracting. *J. ACM* 16:264–285. https://api.semanticscholar.org/CorpusID:1177942.

Kavita Ganesan. 2018. Rouge 2.0: Updated and improved measures for evaluation of summarization tasks. https://arxiv.org/abs/1803.01937.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. https://arxiv.org/abs/1910.13461.

Hans Peter Luhn. 1958. The automatic creation of literature abstracts. *IBM J. Res. Dev.* 2:159–165. https://api.semanticscholar.org/CorpusID:15475171.

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

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In Lluís Màrquez, Chris Callison-Burch, and Jian Su, editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, pages 379–389. https://doi.org/10.18653/v1/D15-1044.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. https://arxiv.org/abs/1704.04368.

Sajad Sotudeh, Hanieh Deilamsalehy, Franck Dernoncourt, and Nazli Goharian. 2023. Curriculum-guided abstractive summarization. https://arxiv.org/abs/2302.01342.

Ilya Sutskever, Oriol Vinyals, and Quoc Le. 2014. Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems* 4.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020a. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. https://arxiv.org/abs/1912.08777.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020b. Bertscore: Evaluating text generation with bert. https://arxiv.org/abs/1904.09675.