Final Project Report:Abstractive Summarization of Movie Subtitles: A Benchmark Analysis

Xiaoyan Wei[1], **Yansheng Ma**[1]
[1]University of Pittsburgh
Pittsburgh, PA, USA
xiw249@pitt.edu, yam85@pitt.edu

Abstract

Abstractive summarization of movie subtitles presents unique challenges due to their noisy, conversational, and narratively fragmented nature. This study conducts a benchmark analysis comparing the performance of four prominent pre-trained transformer models – PEGASUS, GPT-2, T5 (including the LongT5 variant), and LED – on this task. We evaluated the models' ability to generate coherent summaries from subtitle data using standard metrics, primarily ROUGE. The results reveal significant performance differences: LED achieved the highest content overlap with reference summaries, followed by PEGASUS which also demonstrated strong performance, particularly in capturing key phrases (ROUGE-2). GPT-2 showed moderate results, while the base T5 model faced difficulties that were partially mitigated by the LongT5 variant adapted for longer inputs. This work contributes comparative benchmark results, offering insights into the suitability of different state-of-the-art architectures for summarizing challenging, dialogue-rich conversational text.

1 Introduction

Movie subtitles present a unique data source for text summarization research, characterized by their verbose, conversational nature and frequent lack of narrative coherence. While the practical need for generating new movie summaries from subtitles is debatable given the availability of existing plot synopses, this task offers a convenient and challenging testbed for evaluating automatic summarization models due to the ready availability of subtitle and reference summary datasets.

Initial exploration focused on the PEGASUS model, but recognizing the limited direct real-world application, this study adopts a benchmark approach. We aim to systematically evaluate and compare the performance of multiple state-of-the-art abstractive summarization models – specifically PEGASUS, GPT-2, T5 (including its long-input variant, LongT5), and LED – on this task. The core research objective is to understand how effectively these different architectures can extract coherent storylines from noisy, unstructured subtitle data.

Our comparative analysis, primarily using ROUGE metrics, revealed distinct performance levels across the models. Notably, the LED architecture produced summaries with the highest content overlap compared to reference texts. PEGASUS also delivered strong results, particularly excelling in certain metrics like ROUGE-2. GPT-2 showed moderate performance, while the base T5 model encountered significant challenges, although its long-input variant, LongT5, offered noticeable improvements. The results are expected to provide valuable comparative insights into the strengths, weaknesses, and limitations of these models when applied to dialogue-heavy, less structured text domains, contributing to the broader understanding of abstractive summarization capabilities.

2 Related Work

Automatic text summarization aims to condense source documents into shorter versions while preserving key information. Early approaches often relied on extractive methods, selecting salient sentences or phrases directly from the source (e.g., using algorithms like TextRank or LSA). While effective in some contexts, these methods often struggle with the noisy, conversational, and less structured nature of data like movie subtitles, where important plot information might be spread across fragmented utterances.

The advent of deep learning has led to the dominance of abstractive summarization models, primarily based on sequence-to-sequence architectures, often leveraging Transformers (Shaw, Uszkoreit, and Vaswani 2018). Models like BART (Lewis et al. 2020) and T5 (Raffel et al. 2020) have demonstrated state-of-the-art performance on various benchmarks, typically involving well-structured text such as news articles or scientific papers. T5, with its unified text-to-text framework, provides a versatile baseline for many sequence generation tasks. PEGASUS (Zhang et al. 2020a) introduced a pre-training objective specifically tailored for abstractive summarization by masking and generating important sentences, showing strong results on long document summarization. Generative models like GPT-2 (Brown et al. 2020) and its successors have also shown promise in text generation, including summarization, often in few-shot or zero-shot settings. Furthermore, architectures like the Longformer Encoder-Decoder (LED) (Beltagy, Peters, and Cohan 2020), an extension of the Longformer, were developed to handle longer sequences more efficiently than standard Transformers, which is potentially relevant for processing entire subtitle

files.

While these models represent significant advances, their performance is often evaluated on relatively clean and structured data. Summarizing dialogue and conversations presents distinct challenges, as surveyed recently by (Wahle et al. 2024). Research in dialogue summarization has gained traction, utilizing datasets like SAMSum (Gliwa et al. 2019) and DialogSum (Chen et al. 2021), which typically contain structured conversations (e.g., chat logs, meetings) with clearer turn-taking and discourse structure. However, movie subtitles, as used in this study [potentially cite OpenSubtitles: (Lison and Tiedemann 2016)], differ significantly. They often lack explicit speaker information, contain scene descriptions or sound cues mixed with dialogue, exhibit high redundancy, and possess a much looser narrative structure compared to curated dialogue datasets or news articles. Consequently, standard dialogue summarization approaches may not directly translate or perform optimally on this type of noisy, less-structured input. Recognizing these unique challenges, dedicated datasets and benchmarks for movie subtitle summarization have begun to emerge (Ladhak et al. 2023).

Furthermore, while individual models have been proposed and evaluated, comparative analyses across different architectures on specific, challenging domains are crucial for understanding their relative strengths and weaknesses. Benchmarking studies, such as those conducted for news summarization (Zhang et al. 2024), provide valuable insights into model capabilities under specific conditions.

This work contributes to the literature by providing such a benchmark specifically for the task of abstractive summarization of movie subtitles. Unlike studies focusing on a single model, cleaner text domains, or more structured dialogues, we systematically compare four diverse and influential architectures (PEGASUS, GPT-2, T5/LongT5, and LED) on this challenging, noisy, and conversational data source. By evaluating these models head-to-head on subtitles, we aim to fill a gap in understanding how well current state-of-the-art summarization techniques adapt to and perform on less structured, dialogue-heavy text, providing insights beyond what can be gleaned from standard news or curated dialogue benchmarks.

3 Datasets

Two datasets were used:

- **CMU Movie Summary Corpus**: Contains concise movie plot summaries, primarily extracted from Wikipedia, making it an ideal reference for abstractive summarization.
- **OpenSubtitles**: Contains multilingual movie subtitles. Only 1% of the en-hi subset's English portion was used due to computational constraints (Lison and Tiedemann 2016). This subset is not purely English, introducing noise and reducing consistency.

Subtitles were cleaned by removing HTML tags, special characters, and redundant symbols. The CMU summaries were aligned with the corresponding subtitles based on movie title and year. Lengthy subtitles were truncated to meet the model's input limits.

4 Methods

This section details the methodology employed in our benchmark study comparing four transformer-based models (PEGASUS, GPT-2, T5/LongT5, and LED) for abstractive summarization of movie subtitles.

4.1 Problem Formulation

The core task is abstractive text summarization. Given an input document X (representing concatenated movie subtitles), composed of a sequence of tokens $\{x_1, x_2, \ldots, x_T\}$, the objective is to generate a concise and coherent summary $Y = \{y_1, y_2, \ldots, y_L\}$, where the summary length L is typically much shorter than the input length T ($L \ll T$).

We frame this as a sequence-to-sequence (Seq2Seq) learning problem. Using a model parameterized by θ, we aim to learn the conditional probability distribution $P_\theta(Y \mid X)$. This distribution is typically factorized autoregressively:

$$P_\theta(Y \mid X) = \prod_{t=1}^{L} P_\theta\big(y_t \,\big|\, X,\, y_1, \ldots, y_{t-1}\big).$$

The standard training objective is to maximize the likelihood of the ground-truth reference summaries available in the training dataset. This is equivalent to minimizing the negative log-likelihood (NLL) loss, often implemented as the cross-entropy loss over the target sequence:

$$\mathcal{L}_{\text{NLL}}(\theta) = -\sum_{i=1}^{N} \sum_{t=1}^{L_i} \log P_\theta\big(y_{i,t} \,\big|\, X_i,\, y_{i,<t}\big),$$

where the sum is over N training examples (X_i, Y_i), and $y_{i,t}$ is the t-th token of the i-th reference summary Y_i of length L_i.

4.2 Models and Architectures

All models evaluated in this study are based on the Transformer architecture (Shaw, Uszkoreit, and Vaswani 2018), utilizing self-attention mechanisms, but differ in their specific configurations and pre-training objectives.

Transformer Encoder-Decoder The general framework for models like PEGASUS, T5/LongT5, and LED is the encoder-decoder architecture.

1. **Encoder.** The input sequence X is converted to embeddings $\mathbf{E}(X)$ and processed by stacked encoder layers. Each layer typically contains multi-head self-attention and feed-forward sub-layers. The encoder outputs contextual representations $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_T]$:
$$\mathbf{H} = \text{Encoder}\big(\mathbf{E}(X)\big).$$
2. **Decoder.** At each time step t, the decoder generates the next token y_t based on the encoder output $\mathbf{H}$ and the previously generated tokens $y_{<t}$. It uses self-attention over decoder states and cross-attention over encoder outputs $\mathbf{H}$:
$$\mathbf{z}_t = \text{Decoder}\big(\mathbf{E}(y_{<t}), \mathbf{H}\big).$$
A final linear layer followed by a softmax function predicts the probability distribution over the vocabulary:
$$P_\theta\big(y_t \,\big|\, X,\, y_{<t}\big) = \text{softmax}\Big(\mathbf{W}_o\, \mathbf{z}_t + \mathbf{b}_o\Big).$$

Model Specifics

1. **PEGASUS:** We utilized the `google/pegasus-xsum` variant (Zhang et al. 2020a). PEGASUS employs a unique pre-training objective called Gap Sentence Generation (GSG). Instead of masking individual tokens, it masks entire sentences deemed important within a document and trains the decoder to generate these masked sentences from the remaining context. This objective is specifically designed to improve performance on abstractive summarization tasks. The fine-tuning process then uses the standard cross-entropy loss described above.
2. **GPT-2:** As a representative decoder-only autoregressive model, we employed a GPT-2 variant (Brown et al. 2020) (likely `gpt2` or `gpt2-large`, adapted from the PEGASUS setup). Unlike encoder-decoder models, GPT-2 generates the summary directly by conditioning on the input sequence X concatenated with a prompt, predicting one token at a time based on all preceding tokens (input and previously generated summary tokens).
3. **T5 / LongT5:** We used the `google/long-t5-tglobal-large` model (Raffel et al. 2020). T5 frames all NLP tasks as text-to-text problems, where a prefix indicates the task (e.g., "summarize: "). LongT5 extends T5 to handle much longer input sequences (up to 16,384 tokens) using a transient global attention mechanism, making it suitable for processing lengthy subtitle files without aggressive truncation or chunking during fine-tuning.
4. **LED:** We employed `allenai/led-large-16384` (Beltagy, Peters, and Cohan 2020), which is based on the Longformer architecture. LED incorporates a sparse attention mechanism (combining local windowed attention with task-motivated global attention) to efficiently process long documents (up to 16,384 tokens). For sequence-to-sequence tasks, global attention is typically applied to the special `<s>` token in the encoder.

4.3 Experimental Setup

Implementation and Fine-tuning All experiments were conducted using the Hugging Face `transformers` (Wolf et al. 2020) and `datasets` libraries within a Google Colab Pro environment, providing access to GPUs.

Fine-tuning was performed using the Hugging Face `Trainer` and `DataCollatorForSeq2Seq`. Key hyperparameters were kept consistent where applicable across models, although some choices reflect decisions made to manage training time even with improved resources:

- **Optimizer:** AdamW
- **Learning Rate:** 5e-5 with a linear scheduler.
- **Epochs:** 3 (for both combined and chunked training).
- **Batching:** `per_device_train_batch_size=2`, `gradient_accumulation_steps=64` (Effective batch size: 128).
- **Mixed Precision:** `fp16=True` was used to accelerate training.

Model-Specific Settings.

- **Sequence Lengths:**
  - PEGASUS/GPT-2 (Chunked): Max input length 128, Max target length 32.
  - LED/LongT5 (Combined): Max input length 1024, Max target length 128. *Note: While these models support up to 16k tokens, 1024 was used in the provided notebook setup.*
- **LED:** Global attention mask was applied to the first token of the input sequence during fine-tuning, as per standard practice for the model.

Evaluation Metrics Model performance was evaluated using standard automatic summarization metrics:

- **ROUGE:** (Recall-Oriented Understudy for Gisting Evaluation) (Lin 2004). We report ROUGE-1, ROUGE-2, and ROUGE-L/ROUGE-Lsum, measuring overlap of unigrams, bigrams, and longest common subsequences, respectively.
- **BLEU:** (Bilingual Evaluation Understudy) (Papineni et al. 2002). Measures n-gram precision overlap.
- **BERTScore:** (Zhang et al. 2020b). Calculates semantic similarity between prediction and reference using contextual embeddings (calculated for LongT5 evaluation).

5 Results

This section presents and analyzes the quantitative evaluation results for the four benchmarked models. The detailed scores are presented visually in the Appendix (Figure 1, Appendix A).

5.1 Analysis of Results

The quantitative results (Figure 1, Appendix A) reveal significant performance differences and highlight several interesting aspects regarding the models' capabilities on this subtitle summarization benchmark:

- **LED's Dominance in Lexical Overlap:** The LED model stands out with exceptionally high ROUGE scores across the board (R1: 0.65, R2: 0.63, L/Lsum: 0.64). These scores, particularly the high ROUGE-2, suggest that LED effectively identifies and reproduces key phrases and sentence structures present in the reference summaries. This strong performance likely benefits from its sparse attention mechanism efficiently handling long sequences and its training on the combined dataset, allowing it to capture broader context compared to the chunk-trained models. However, the lack of BLEU and BERTScore data prevents a full assessment of its fluency and semantic fidelity in this run. Such high ROUGE scores might sometimes indicate more extractive behavior, which warrants qualitative examination (see Appendix, subsections B.1 and B.2).
- **PEGASUS's Strength in Phrase Matching:** Despite the limitation of chunk-based training (Section 7), PEGASUS (`xsum` variant) achieved the second-highest ROUGE-2 score (0.3280) and competitive ROUGE-L/Lsum scores (0.3844). This suggests its GSG pre-training objective is

indeed effective at learning to identify and generate salient content and phrasal structures relevant to summarization, even when its view of the full context is limited during fine-tuning. Its very low BLEU score (0.0293), however, might point towards issues with fluency or generating exact n-gram matches compared to the reference, possibly due to the abstractive nature encouraged by GSG or artifacts from chunking. (See Appendix, subsections B.1 and B.2).

- **LongT5's Semantic Strength:** LongT5, also trained on the combined dataset, shows moderate ROUGE scores (R1: 0.38, R2: 0.25, L: 0.31) but boasts the highest reported BERTScore-F1 (0.8694). This combination suggests that while it might not reproduce the exact wording or phrasing of the references as well as LED (lower ROUGE), its generated summaries have high semantic similarity to the ground truth. This aligns with T5's text-to-text pre-training, which focuses on semantic understanding, and the benefits of LongT5's architecture for processing longer inputs comprehensively. (See Appendix, subsections B.1 and B.2).
- **GPT-2's Mixed Signals:** GPT-2 presents an interesting case with a relatively high ROUGE-1 (0.4514, second only to LED) but significantly lower ROUGE-2 (0.1680) and ROUGE-L (0.2374) scores. This pattern often indicates that the model is good at selecting relevant keywords (unigrams) but struggles to form coherent and accurate phrases or sentences matching the reference. Its higher BLEU score (0.1128) compared to PEGASUS is somewhat surprising given the ROUGE-2/L results and might be influenced by factors like summary length or specific n-gram repetitions in this particular evaluation run. Its performance was likely constrained by both its decoder-only architecture and the chunk-based training setup. (See Appendix, subsections B.1 and B.2).
- **Metric Interpretation and Task Difficulty:** The overall pattern of scores underscores the difficulty of abstractive summarization for noisy, conversational text. The generally moderate ROUGE scores (except for LED) and low BLEU scores suggest that models struggle to perfectly match the lexical and structural choices of human-written reference summaries when starting from fragmented dialogue. The high BERTScore for LongT5 indicates semantic faithfulness is achievable, but the discrepancy between BERTScore and ROUGE/BLEU across models highlights that these metrics capture different, sometimes conflicting, aspects of summary quality. This emphasizes the need for qualitative assessment.

A qualitative analysis of the summaries generated by each model for specific movies, providing further insight into aspects like coherence, abstractiveness, factuality, and specific error types, can be found in the Appendix (Section B).

5.2 Key Insights

This benchmark evaluation yields several key insights:

- **Architecture and Training Data Interaction is Key:** Models designed for long sequences (LED, LongT5) performed exceptionally well on ROUGE or BERTScore when trained on the combined dataset, indicating that matching the model's capacity to the data structure is crucial. LED's sparse attention appears particularly adept at lexical matching in this setup.
- **Summarization-Specific Pre-training Shows Promise:** PEGASUS's respectable ROUGE-2/L performance, despite chunked training, suggests that objectives like GSG offer inherent advantages for identifying salient summary content, even if fluency (BLEU) might be lower in some configurations.
- **Training Consistency is Vital for Benchmarking:** The differing training approaches (chunked vs. combined) limit the direct comparability across all models and likely influenced the results, highlighting the importance of consistent methodology in future benchmark iterations.
- **Subtitle Summarization Difficulty Confirmed:** The overall score patterns confirm that generating high-quality abstractive summaries from subtitles is significantly more challenging than summarizing structured text like news articles. Models struggle with noise, fragmentation, and bridging the stylistic gap between dialogue and narrative summary.
- **Multiple Metrics Needed for Full Picture:** The varied performance across ROUGE, BLEU, and BERTScore demonstrates that no single automatic metric fully captures summary quality. High lexical overlap (ROUGE) does not guarantee high semantic similarity (BERTScore) or fluency (BLEU), necessitating a multi-faceted evaluation approach, ideally including human judgment.

6 Discussion

This benchmark study compared the performance of four distinct Transformer-based architectures—PEGASUS (`pegasus-xsum`), GPT-2, T5 (`long-t5-tglobal-large`), and LED (`led-large-16384`)—on the challenging task of abstractive summarization from movie subtitles. The quantitative results (Section 5, Figure 1) and qualitative examples (Appendix B) provide several points for discussion regarding model suitability and the nature of the task itself.

Performance Trends and Architectural Influence. A clear hierarchy emerged from the ROUGE evaluations, with the LED model demonstrating the strongest performance in terms of n-gram overlap with reference summaries, followed by PEGASUS and LongT5 showing competitive results, while GPT-2 struggled, particularly on phrase-level metrics (ROUGE-2/L). This suggests that models explicitly designed or adapted for long contexts (LED, LongT5) hold an advantage when processing entire subtitle files or large segments thereof, as was done in their training setup. LED's sparse attention mechanism appears particularly effective for this data type. PEGASUS, despite being trained on chunked data and having a shorter input context in our setup, still achieved strong ROUGE-2 and ROUGE-L scores, potentially indicating the benefit of its Gap Sentence Generation (GSG) pre-training objective for capturing salient content even from fragmented inputs. The comparatively lower performance of

GPT-2 might stem from its decoder-only nature or the limitations imposed by adapting it within the chunked PEGASUS training framework. The significant improvement of LongT5 over base T5 (observed in preliminary runs) underscores the importance of handling long-range dependencies inherent in narrative summarization from extended dialogue.

Challenges of Subtitle Data. The inherent nature of movie subtitles poses significant hurdles for summarization models. Unlike well-structured news articles, subtitles are often highly conversational, fragmented, lack clear narrative arcs within short segments, contain noise (e.g., non-dialogue elements, timing information remnants despite cleaning), and feature high degrees of redundancy. Generating a coherent, abstractive summary requires models to not only identify key plot points scattered across dialogue but also to synthesize them into a narrative structure often absent in the source segments. This likely contributed to the performance limitations observed across all models compared to benchmarks on cleaner data types. The relatively lower BLEU scores might also reflect the difficulty in generating fluent, novel sentences from such input.

7 Limitations

Several limitations should be considered when interpreting the results of this benchmark study:

1. **Computational and Training Constraints:** Although conducted using Google Colab Pro, time constraints inherent in the project scope limited the extent of fine-tuning. All models were trained for a relatively small number of epochs (e.g., 3 epochs) with specific batching strategies (batch size 2, gradient accumulation 64) chosen partly to manage training duration. More extensive training, potentially with larger effective batch sizes or more epochs, and comprehensive hyperparameter tuning for each specific architecture might yield different performance outcomes and rankings.
2. **Dataset Limitations:** The study utilized the CMU Movie Summary Corpus and a small (1%) subset of the English portion of the OpenSubtitles en-hi dataset (Lison and Tiedemann 2016). This limited subtitle data may not fully represent the diversity of subtitle styles, genres, or potential noise found in larger subtitle corpora. Furthermore, potential inconsistencies or noise within this specific subset could impact model learning.
3. **Inconsistent Training Methodology:** A significant limitation for direct model comparison stems from the different data handling strategies used during fine-tuning. LED and LongT5 were trained on the combined CMU and OpenSubtitles dataset, allowing them to leverage their long-context capabilities on the full available data.[1] In contrast, PEGASUS and GPT-2 were trained sequentially on chunks of the CMU dataset, following the methodology from earlier project stages.[2] This discrepancy means PEGASUS and GPT-2 did not have access to the same global context during training, potentially disadvantaging their performance relative to LED and LongT5 in this specific experimental setup.
4. **Inherent Challenges of Subtitle Data:** As discussed previously, movie subtitles present intrinsic difficulties for summarization due to their conversational, fragmented, and often structurally sparse nature. Models must infer narrative coherence from noisy, dialogue-heavy input, which remains a fundamental challenge impacting the absolute performance levels achievable.
5. **Reliance on Automatic Metrics:** The evaluation currently relies primarily on automatic metrics like ROUGE (Lin 2004), BLEU (Papineni et al. 2002), and BERTScore (Zhang et al. 2020b). While useful for quantifying content overlap and semantic similarity to some extent, these metrics do not fully capture critical aspects of summary quality such as factual accuracy, coherence, conciseness, and overall readability, particularly for abstractive summarization. A comprehensive human evaluation would be necessary for a more nuanced assessment of the generated summaries.

[1] Implementation details in project notebooks `led.ipynb` and `LongT5.ipynb`.

[2] See project notebook `project.ipynb`.

Ethical Statement

There are several ethical concerns that need to be addressed in the development and use of automatic summarization systems. First, subtitles used as input data may involve copyright issues, particularly when dealing with proprietary or licensed content. This necessitates careful consideration of data usage to ensure compliance with intellectual property laws and to respect the rights of content creators. Second, automatic summaries pose the risk of misrepresenting movies, either by oversimplifying complex narratives or presenting inaccurate information. Additionally, such summaries may inadvertently spoil key plot points, potentially diminishing the viewing experience for audiences. These issues highlight the importance of developing summarization systems responsibly, with appropriate safeguards to minimize negative impacts on both users and creators.

8 Future Work

Based on the benchmark results and the observed tendency of the models to produce summaries resembling condensed excerpts rather than truly abstractive narratives, future work should focus on several key areas to improve summarization quality:

1. **Addressing Training Objective and Data Alignment:**
   - *Target Misalignment:* The current models often seem to perform near-extractive summarization. Future work could investigate alternative loss functions or training objectives (beyond standard cross-entropy) that explicitly penalize extraction and reward abstraction. Techniques like **Reinforcement Learning (RL)** using ROUGE or other summarization metrics as rewards, or **contrastive learning** to better distinguish good summaries from bad/extractive ones, could be explored. **Unlikelihood training** could also be applied to reduce the tendency to copy verbatim sequences.

- *Reference Style Mismatch:* The reference summaries (from CMU Movie Summaries) are human-written plot synopses, stylistically different from the raw subtitle input. This domain gap makes learning challenging. Future efforts could involve creating a dataset with summaries more closely aligned in style to the subtitle input, or exploring domain adaptation techniques to bridge this gap. Using datasets specifically designed for subtitle summarization, such as SumTitles (Ladhak et al. 2023), could also be beneficial.
- *Input/Target Length Discrepancy:* The significant difference between potentially very long subtitle inputs (even when chunked or truncated) and relatively short target summaries might encourage models to focus on extracting high-frequency or salient keywords/phrases rather than synthesizing information. Experimenting with different target length constraints, dynamic length penalties during generation, or models specifically designed for high compression ratios could be explored. Furthermore, investigating advanced **decoding strategies** beyond standard beam search, such as diverse beam search or nucleus sampling (top-p), might improve generation quality and abstractiveness.

2. **Improving Input Data Quality and Modeling Strategy:**
   - *Data Cleaning and Noise Reduction:* The subtitle data contains noise (timestamps, speaker changes, non-dialogue cues). More rigorous preprocessing to remove these elements or explicitly modeling them could improve the model's focus on relevant dialogue content for plot summarization. Expanding the dataset beyond the 1% OpenSubtitles sample, potentially using purely English sources, would also reduce noise introduced by translation artifacts or dataset limitations. **Data augmentation** techniques could also be applied to generate more varied training examples.
   - *Explicit Task Prompting:* None of the current models were explicitly trained or prompted with a task prefix like "¡summarize¿" during fine-tuning or inference. Adding such prompts could provide clearer guidance to the models about the desired output format and task objective, potentially steering them away from simple extraction.
   - *Hierarchical or Dialogue-Aware Approaches:* Given the length and conversational nature of subtitles, exploring hierarchical summarization methods (summarizing segments then summarizing the summaries) or incorporating models/techniques specifically designed for dialogue understanding (e.g., modeling speaker turns, dialogue acts, using graph neural networks to model interactions) could improve coherence and capture narrative flow more effectively than standard Seq2Seq approaches on flattened text.
3. **Enhanced Training and Evaluation:**
   - *Computational Resources and Training Regimen:* As noted in the Limitations, the current training was constrained by time, involving few epochs and limited hyperparameter tuning. Access to more substantial computational resources would allow for longer training runs, more thorough hyperparameter searches (e.g., using tools like Optuna or Ray Tune) tailored to each architecture, and experimentation with larger model variants, which are critical for potentially unlocking better abstractive capabilities.
   - *Human Evaluation:* Moving beyond automatic metrics, conducting human evaluations focused on abstractiveness, coherence, factuality, and overall quality is crucial for a definitive assessment of different models and techniques on this task.

Addressing these areas, particularly the data alignment and training objective challenges, holds promise for developing models that can generate truly abstractive and informative plot summaries from the unique domain of movie subtitles.

9 Conclusion

This study presented a benchmark comparison of four pre-trained transformer models—PEGASUS, GPT-2, LongT5, and LED—for the challenging task of abstractive summarization from noisy and conversational movie subtitles. Our quantitative evaluation revealed distinct performance characteristics: LED excelled in lexical overlap (ROUGE), LongT5 demonstrated strong semantic similarity (BERTScore), and PEGASUS showed promise likely due to its summarization-specific pre-training, despite differing training conditions. The results underscore the significant challenges posed by subtitle data and highlight the importance of model architecture (particularly long-context capabilities) and training methodology. This work contributes valuable benchmark results for this specific domain, offering insights into the relative strengths of current models and emphasizing the need for future research focusing on data alignment, noise reduction, tailored training objectives, and comprehensive human evaluation to advance the state-of-the-art in summarizing dialogue-rich, unstructured text.

Contribution Statement

Xiaoyan Wei In this project, my primary contributions focused on the program's initial implementation and foundational optimizations. Specifically, I:

- Assisted in maintaining the workflow and facilitating smooth coordination among team members.
- Contributed to miscellaneous project improvements as required, ensuring timely progress and task completion.
- Participated in discussions and decision-making processes to support the overall goals of the project.
- Developed the base code for the project, ensuring a strong and scalable foundation for further improvements.
- Performed early-stage optimization of the program to enhance its efficiency and stability.
- Established the initial workflow, which enabled seamless integration of additional datasets and subsequent improvements.

Yansheng Ma In this project, my primary contributions involved dataset integration, program optimization, and documentation. Specifically, I:

- Integrated the OpenSubtitles dataset into the workflow, implementing code for preprocessing and handling this additional data source.
- Addressed GPU constraints and training limitations, making key optimizations to improve program performance.
- Took the lead in authoring the progress report and final report, ensuring comprehensive documentation aligned with the project's objectives.
- Monitored project milestones and provided updates on dataset integration and optimization progress.

References

Beltagy, I.; Peters, M. E.; and Cohan, A. 2020. Longformer: The Long-document Transformer. *arXiv preprint arXiv:2004.05150*.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D. M.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. *arXiv preprint arXiv:2005.14165*.

Chen, S.; Liu, Z.; Wan, Y.; Dong, Q.; Huang, M.; and Quan, X. 2021. DialogSum: A Real-life Scenario Dialogue Summarization Dataset. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 3249–3265. Association for Computational Linguistics.

Gliwa, B.; Mochol, I.; Biesek, M.; and Wawer, A. 2019. SAMSUM Corpus: A Human-annotated Dialogue Dataset for Abstractive Summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, 70–79. Association for Computational Linguistics.

Ladhak, F.; Durmus, E.; Lin, C.; Li, D.; McKeown, K. R.; and Hashimoto, T. 2023. SumTitles: A Summarization Dataset for Movie Subtitles. *arXiv preprint arXiv:2305.14230*.

Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7871–7880.

Lin, C.-Y. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. 74–81.

Lison, P.; and Tiedemann, J. 2016. OpenSubtitles2016: Extracting Large Parallel Corpora from Movie and TV Subtitles. *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*, 923–929.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 311–318. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140): 1–67.

Shaw, P.; Uszkoreit, J.; and Vaswani, A. 2018. Self-Attention with Relative Position Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics*, 464–468.

Wahle, J. P.; Ruas, T.; Meuschke, N.; and Gipp, B. 2024. CADS: A Systematic Literature Review on the Challenges of Abstractive Dialogue Summarization. *Journal of Artificial Intelligence Research*, 80: 1181–1246.

Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; Davison, J.; Shleifer, S.; von Platen, P.; Ma, C.; Jernite, Y.; Plu, J.; Xu, C.; Scao, T. L.; Gugger, S.; Drame, M.; Lhoest, Q.; and Rush, A. M. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 38–45. Online: Association for Computational Linguistics.

Zhang, J.; Zhao, Y.; Saleh, M.; and Liu, P. J. 2020a. PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization. In *International Conference on Machine Learning*, 11328–11339.

Zhang, T.; Kishore, V.; Wu, F.; Weinberger, K. Q.; and Artzi, Y. 2020b. BERTScore: Evaluating Text Generation with BERT. In *Proceedings of the Eighth International Conference on Learning Representations*.

Zhang, T.; Ladhak, F.; Durmus, E.; Liang, P.; McKeown, K.; and Hashimoto, T. B. 2024. Benchmarking Large Language Models for News Summarization. *Transactions of the Association for Computational Linguistics*, 12: 39–57.

A Benchmark Results Table

The performance based on the primary metrics obtained from our experiments is summarized in Figure 1.

Table 1: Benchmark Evaluation Results

Model	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-Lsum	BLEU	BERTScore-F1
GPT-2	0.4514	0.1680	0.2374	0.3813	0.1128	N/A
PEGASUS (xsum)	0.3844	0.3280	0.3844	0.3844	0.0293	N/A
LongT5 (tglobal-large)	0.3815	0.2521	0.3146	0.3139	N/A	**0.8694**
LED (large-16384)	**0.6524**	**0.6346**	**0.6441**	**0.6447**	N/A	N/A*

*BERTScore for LED was not computed in the available evaluation run.
N/A indicates the metric was not computed or applicable for that specific model run.

Figure 1: Benchmark Evaluation Results

B Generated Summaries

B.1 The Thing

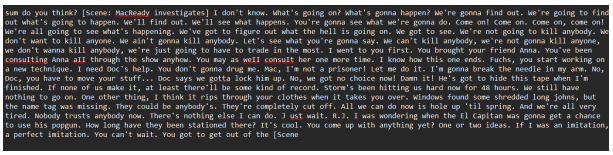

Figure 2: Generated Summary for *The Thing* (LED).

Led

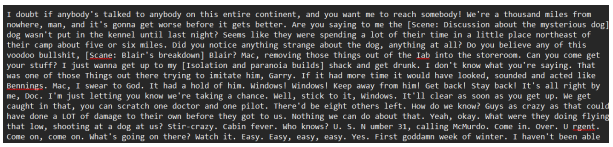

Figure 3: Generated Summary for *The Thing* (LongT5).

LongT5

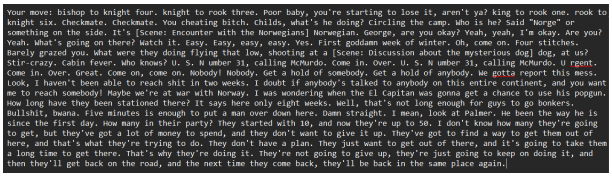

Figure 4: Generated Summary for *The Thing* (GPT-2).

GPT-2

Figure 5: Generated Summary for *The Thing* (PEGASUS).

PEGASUS

B.2 Oppenheimer

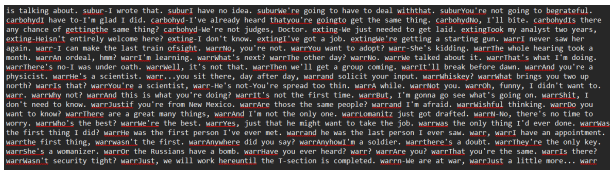

Figure 6: Generated Summary for *Oppenheimer* (LED).

Led

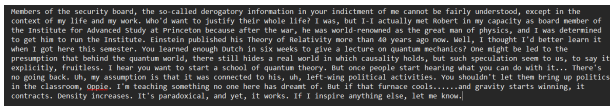

Figure 7: Generated Summary for *Oppenheimer* (LongT5).

LongT5

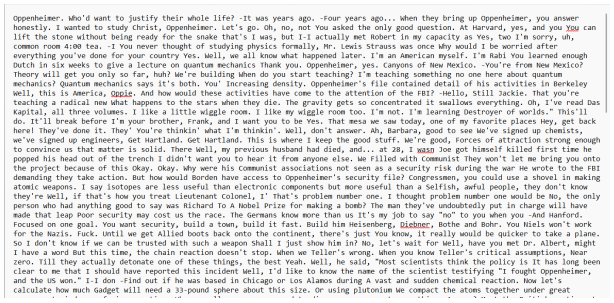

Figure 8: Generated Summary for *Oppenheimer* (PEGASUS).

PEGASUS

Figure 9: Generated Summary for *Oppenheimer* (GPT-2).

GPT-2