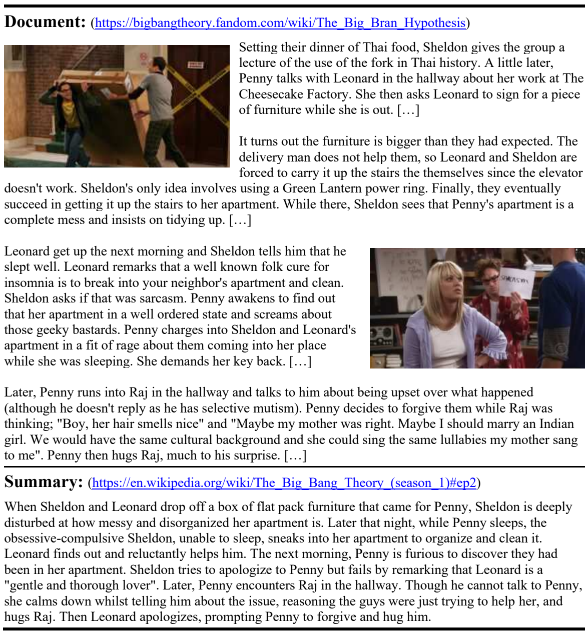
NarraSum: A Large-Scale Dataset for Abstractive Narrative Summarization

Chao Zhao[1] **Faeze Brahman**[2,3] **Kaiqiang Song**[4]
Wenlin Yao[4] **Dian Yu**[4] **Snigdha Chaturvedi** [1]
{zhaochao, snigdha}@cs.unc.edu faezeb@allenai.org
{riversong, wenlinyao, yudian}@global.tencent.com
[1] UNC Chapel Hill [2] Allen Institute for AI [3] University of Washington [4] Tencent AI Lab

Abstract

Narrative summarization aims to produce a distilled version of a narrative to describe its most salient events and characters. Summarizing a narrative is challenging as it requires an understanding of event causality and character behaviors. To encourage research in this direction, we propose NarraSum, a large-scale narrative summarization dataset. It contains 122K narrative documents, which are collected from plot descriptions of movies and TV episodes with diverse genres, and their corresponding abstractive summaries. Experiments show that there is a large performance gap between humans and the state-of-the-art summarization models on NarraSum. We hope that this dataset will promote future research in summarization, as well as broader studies of natural language understanding and generation. The dataset is available at `https://github.com/zhaochaocs/narrasum`.



Document: (https://bigbangtheory.fandom.com/wiki/The_Big_Bran_Hypothesis)

Setting their dinner of Thai food, Sheldon gives the group a lecture of the use of the fork in Thai history. A little later, Penny talks with Leonard in the hallway about her work at The Cheesecake Factory. She then asks Leonard to sign for a piece of furniture while she is out. […]

It turns out the furniture is bigger than they had expected. The delivery man does not help them, so Leonard and Sheldon are forced to carry it up the stairs the themselves since the elevator doesn't work. Sheldon's only idea involves using a Green Lantern power ring. Finally, they eventually succeed in getting it up the stairs to her apartment. While there, Sheldon sees that Penny's apartment is a complete mess and insists on tidying up. […]

Leonard get up the next morning and Sheldon tells him that he slept well. Leonard remarks that a well known folk cure for insomnia is to break into your neighbor's apartment and clean. Sheldon asks if that was sarcasm. Penny awakens to find out that her apartment in a well ordered state and screams about those geeky bastards. Penny charges into Sheldon and Leonard's apartment in a fit of rage about them coming into her place while she was sleeping. She demands her key back. […]

Later, Penny runs into Raj in the hallway and talks to him about being upset over what happened (although he doesn't reply as he has selective mutism). Penny decides to forgive them while Raj was thinking; "Boy, her hair smells nice" and "Maybe my mother was right. Maybe I should marry an Indian girl. We would have the same cultural background and she could sing the same lullabies my mother sang to me". Penny then hugs Raj, much to his surprise. […]

Summary: (https://en.wikipedia.org/wiki/The_Big_Bang_Theory_(season_1)#ep2)

When Sheldon and Leonard drop off a box of flat pack furniture that came for Penny, Sheldon is deeply disturbed at how messy and disorganized her apartment is. Later that night, while Penny sleeps, the obsessive-compulsive Sheldon, unable to sleep, sneaks into her apartment to organize and clean it. Leonard finds out and reluctantly helps him. The next morning, Penny is furious to discover they had been in her apartment. Sheldon tries to apologize to Penny but fails by remarking that Leonard is a "gentle and thorough lover". Later, Penny encounters Raj in the hallway. Though he cannot talk to Penny, she calms down whilst telling him about the issue, reasoning the guys were just trying to help her, and hugs Raj. Then Leonard apologizes, prompting Penny to forgive and hug him.

Figure 1: Example of the narrative summarization task. The input is a narrative text (denoted by "Document", **pictures are not included**), and the output is a summary containing its salient events and characters.

1 Introduction

A narrative is a story (e.g., a novel or a movie) composed of events and characters (Prince, 1973). *Narrative summarization* aims to produce a distilled version of a narrative, either extractively or abstractively, to contain its most salient events and major characters (Lehnert, 1981). This ability is especially crucial for the understanding of narratives, and in general, the understanding of human behaviors and beliefs (Piper et al., 2021). Practically, a summary of a narrative can enable a reader to quickly discern the key points, which is useful in real-world scenarios such as content recommendations and advertisements.

While text summarization has been explored for over decades, most existing studies focus on summarizing news (Consortium and Company, 2008; Nallapati et al., 2016; Narayan et al., 2018a) or structured documents (e.g., scientific papers (Gidiotis and Tsoumakas, 2019; Cohan et al., 2018)). These documents have specific writing styles. For instance, news is organized such that the first few sentences convey the most important information (Hicks et al., 2016). Scientific papers usually follow a standard structure with a few sections contributing the most to the summary (Gidiotis and Tsoumakas, 2020). It has been demonstrated that many summarization models, including recent ones, heavily rely on these structural clues (Kedzie et al., 2018; Zhong et al., 2019; Zhao et al., 2022a). However, a typical narrative does not contain such structural cues. This suggests that a narrative summarization model has to understand the entire narrative to identify the salient events and characters. While some recent summarization tasks also require understanding an entire document, they focus on conversational domains such as dialogues (Gliwa et al., 2019), emails (Zhang et al., 2021a), and meetings (Zhong et al., 2021). Narratives are

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 182 - 197
December 7-11, 2022 ©2022 Association for Computational Linguistics

different from those genres in nature and are understudied.

Understanding an entire narrative faces unique challenges. A narrative organizes the story into a sequence of events (i.e., plot) in a chronological and causal order (Forster, 1985). Events unfold due to the actions of characters and other event participants, or external forces in stories (Mani, 2012). To identify the salient events, a model needs to understand both **plot** and **characters**. From the plot's perspective, the model needs to understand the causal and temporal relationships between events, as well as how the plot develops from the beginning to the end (Freytag, 1908). From the character's perspective, the model needs to understand the characters' profiles (e.g., personalities, roles, and interpersonal relationships), and how their desires and actions drive the story forward.

Figure 1 illustrates the importance of understanding the entire narrative for summarization. In this example, the main event is "*Sheldon cleans Penny's apartment and gets Leonard in trouble*", which is included in the summary. The side event "*Penny speaks to Raj and forgives Leonard*" is also included since it is the consequence and ending of the main event. Whereas, *"Sheldon gives a lecture of fork"* is not included as it does not impact the development of the plot. Besides the main events, the summary also explains Sheldon's motivation to clean the apartment.

A large-scale high-quality dataset is essential to promote research on this topic. Unfortunately, different from other domains, such as news and scientific papers, where the document and summary can be found from the same data source, narrative documents and their corresponding summaries are usually spread in separate sources. Previous studies collect document-summary pairs of narrative by either creating summaries manually (Ouyang et al., 2017) or matching titles between documents and summaries followed by a manual inspection (Ladhak et al., 2020; Kryściński et al., 2021), making it challenging to enlarge the resulting datasets.

In this work we propose an automatic data construction framework to build a narrative summarization dataset with both large scale and high quality. Specifically, we first collect narratives from plot descriptions of movies or TV episodes through online resources. We choose the plot description because it describes the overall narrative of the movie or TV episode, including the story arcs and major characters. This source is also widely used in narrative-related studies (Linebarger and Piotrowski, 2009; Bamman et al., 2013; Papalampidi et al., 2019; Xiong et al., 2019). After data collection, we build an **align-and-verify** pipeline to automatically align plot descriptions of the same movie or TV episodes from different sources. Finally, we construct document-summary pairs by treating the long plot description as the document to be summarized and the shorter one (of the same movie or TV episode) as the corresponding summary. After filtering out low-quality document-summary pairs, we build **NarraSum**, a large-scale dataset that contains around 122K **narra**tive document-**sum**mary pairs in English. Our data construction framework is generic and thus can potentially be applied to other languages as well.

To gauge the feasibility of NarraSum for the narrative summarization task, we explore different characteristics of this dataset. We observe that compared with other summarization datasets, the narratives in NarraSum are of diverse genres, and the summaries are more abstractive and of varying lengths. Furthermore, rather than focusing on a particular part of the document (as in other summarization datasets), the summaries in NarraSum are designed to cover the entire narratives. It brings new challenges to current summarization methods.

We investigate the performance of several strong baselines and state-of-the-art summarization models on NarraSum. Results show that there is a large gap between human and machine performance in various dimensions, demonstrating that narrative summarization is a challenging task.

The contributions of this paper are four-fold:

- We propose an automatic data construction framework to build a large-scale, high-quality narrative summarization dataset.
- We release the largest narrative summarization dataset to date named NarraSum, with detailed data analysis;
- We investigate the performance of recent summarization models on NarraSum;
- We perform a thorough analysis of the models to point out the challenges and several promising directions.

2 Data Construction

We propose an automatic data construction framework to create a narrative summarization dataset. To this end, we first collect plot descriptions of movies and TV episodes from multiple resources as narratives (Section 2.1). We then align plot descriptions in these resources that refer to the same movie or TV episode (Section 2.2). Finally, we filter the aligned data to construct high-quality document-summary pairs. (Section 2.3). We describe the details of each step as follows.

2.1 Data Collection

We collect plot descriptions of movies and TV episodes from various movie websites and online encyclopedias such as Wikipedia,[1] Fandom,[2] IMDB, [3] TVDB, [4] and TMDB. [5] Note that while we use movie/TV plot descriptions as a source of narrative text, our goal is not to summarize movies and TV episodes themselves but rather to study the task of narrative summarization in a broader sense. Tasks of movie/TV summarization have been addressed by other datasets such as Scriptbase (Gorinski and Lapata, 2015), Screenplay (Papalampidi et al., 2020), and SummScreen (Chen et al., 2022). Those works focus more on summarizing screenplays, which describe the movements, actions, expressions, and dialogue of the characters in a specific structure and format. Compared with general narrative summarization, screenplay summarization presents a different set of challenges such as scene understanding and dialog parsing. Plot descriptions, on the other hand, describe the movie stories from a third-person point of view and present a different set of challenges as we described in Section 1.

To collect plot descriptions, we parse web pages of movies or TV episodes based on HTML tags and use heuristics to match keywords (e.g., *Synopsis*, *Summary*, and *Plot*) that are related to the plot. We then extract the texts under these sections as plot descriptions of the corresponding movies or TV episodes. Besides the plot descriptions, we also collect the meta information of movies or TV episodes such as their title, air date, director(s), and writer(s), which is used for data alignment.

2.2 Data Alignment

After data collection, we align the web pages that are from different websites but refer to the same movie or TV episode. It is a challenging task due to the ambiguity in natural language. For example, a single movie may have different surface forms of the title (e.g., *Avengers 4* and *Avengers: Endgame*), while those with the same title may refer to different movies (e.g., *Bad Company* may refer to fourteen movies.) Similar ambiguity issues arise when aligning air dates or names of crew members. Also, meta-information might be missing or incorrect due to the editing or parsing mistakes of web pages. To address these challenges, we propose an **align-and-verify** pipeline. It first aligns movie or TV episodes via fuzzy meta-information matching, which encourages high recall. Then, we use a verifier with high precision to re-check the aligned pairs and filter out the pairs with low confidence. We describe the details of this pipeline as follows.

During the **alignment** stage, we apply several heuristics for fuzzy meta-information matching. To align movies, we first normalize movie titles by removing non-alphanumeric characters, stopwords, and subtitles. We then collect the movie pairs where the Levenshtein distance between the normalized titles is less than a threshold.[6] Besides the title match, we also require the two movies to have the same air date or a partial overlap on directors or writers when such information is available. The ambiguity in titles of TV episodes is more severe than that of movies. To align TV episodes, we apply similar heuristics and further require the two episodes to belong to the same TV show.

During the **verification** stage, we improve the precision of alignment by comparing the aligned plot descriptions. Specifically, we train a classifier to take as input the concatenation of two plot descriptions to predict if they should be aligned. To train such a classifier, we first build a dataset with balanced positive aligned pairs and negative pairs. The positive pairs are a subset of heuristically aligned pairs where there is an link in one web page (e.g., "External links" in Wikipedia) pointing to the web page of the same movie or TV episode in the other website. Such links are edited by humans and are commonly used in entity linking (Shen et al., 2014). Negative pairs are randomly sampled from different movies of the same movie series or different episodes of the same TV show. Negative pairs sampled by this strategy usually share a similar set of characters and background setting, preventing

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

[2] https://www.fandom.com/.

[3] https://www.imdb.com/.

[4] https://thetvdb.com/.

[5] https://www.themoviedb.org/.

[6] We set the threshold to be $0.2 \times l$, where l is the maximum length of the two titles. All thresholds in this section were chosen by experimenting with different values and manually analyzing the quality of a subset of the data.

the model from relying on surface-level cues to solve the task.

Based on the data sampling method, we collected a large-scale balanced dataset with 60K positive pairs and 60K negative pairs. We then split the dataset into train/validation/test subsets with the ratio of 80%/10%/10%. We train a RoBERTa-base (Liu et al., 2019) classifier on this dataset and it achieves an accuracy of 97.13% on the test set, indicating that this model can serve as a reliable verifier to improve the precision of data alignment. We employ this classifier to further verify the heuristically aligned plot descriptions and filter out those where the predicted log-odds is smaller than 1. Finally, we obtain 2.6 million aligned plot description pairs.

2.3 Document-Summary Pairing

After obtaining the aligned plot description pairs, we regard the longer plot description as the document and the shorter one as the corresponding summary. However, not all pairs are of good quality for summarization. We identify three major issues compromising the quality and remove the relatively low-quality pairs from the final dataset.

First, the summary may contain hallucinated content that might not be included in the document. Similar to (Ladhak et al., 2020), we observe that hallucination is less common in plot description pairs with a noticeable difference in length. We therefore require the length of the summary to be shorter than half of the document to be summarized. We also calculate the semantic matching score between a summary and a document, and then remove the pairs with low scores. We adopt two scores. The first is the Rouge-1 Precision between the summary and the document. The second is the entailment probability between the summary and the document obtained from DocNLI (Yin et al., 2021), a document-level NLI model. We add up the two scores, rank the instances accordingly, and remove the 3% document-summary pairs with the lowest score.

Second, sometimes the content in the shorter plot description is directly copied from the longer plot description. To create an abstractive summarization dataset, we use ROUGE-2 Precision (Lin, 2004) between the document and the summary to reflect whether the content of the summary is copied from the document, and remove the pairs where the ROUGE-2 Precision is larger than 0.5.

Datasets	Domain	Size	L-doc	L-sum	Ratio
CNNDM	News	312K	781	56	13.9
XSum	News	227K	431	20	21.5
arXiv	Sci-Paper	215K	4,938	220	22.4
PubMed	Sci-Paper	133K	3,016	203	14.9
NovelChap	Novel	8K	5,165	372	13.9
BookSum	Novel	12K	5,102	505	10.1
NARRASUM	Movie/TV	122K	786	147	5.3

Table 1: Comparison between NARRASUM and other datasets accroding to the domain, size, document length, summary length, and compression ratio.

Third, a plot description may only describe part of the entire narrative such as a trailer but does not necessarily summarize the narrative. To filter out these cases, we set the minimum length of documents and summaries to make sure that they contain enough information. [7] We also extract oracle extractive summaries from the original document using the method proposed by Liu and Lapata (2019). We remove the instances where less than 30% content of the oracle extractive summaries are from either the first half or the second half of the document.

After applying these filtering strategies, we obtain the final version of NARRASUM. It contains 122K aligned document-summary pairs, which is a high-quality subset (3.8%) of the original aligned pairs. We split the dataset into training (90%), validation (5%), and testing (5%) sets at the title level in order to avoid data leakage and undesirable overlap between training and validation or test sets.

3 Data Analysis

This section provides basic statistics of NARRASUM. We then analyze the dataset in terms of the distribution of salient information and abstractiveness of summaries. Finally, we conduct a human assessment to evaluate the quality of NARRASUM.

3.1 Data Statistics

We compare NARRASUM with six datasets from different domains such as news, scientific papers, and narratives. These include CNN DailyMail (CNNDM) (See et al., 2017), XSum (Narayan et al., 2018b), ArXiv (Cohan et al., 2018), PubMed (Cohan et al., 2018), NovelChapter (Ladhak et al., 2020), and BookSum (Kryściński et al., 2021). The comparison of statistics is shown in Table 1.

[7] For movies, we set the minimum length of documents and summaries as 200 and 100. For TV episodes, we set the minimum length as 100 and 50.

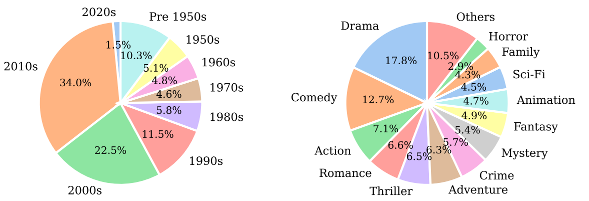


Figure 2: Distribution of production years and genres in NARRASUM.

Datasets	**% of novel n-grams in summary**			
	1-grams	2-grams	3-grams	4-grams
CNN/DM	17.00	53.91	71.98	80.29
XSum	35.76	83.45	95.50	98.49
Pubmed	18.53	48.23	68.28	78.39
NARRASUM	47.78	81.86	94.96	98.00

Table 2: Comparison of novel n-grams between NARRASUM and other summarization datasets.

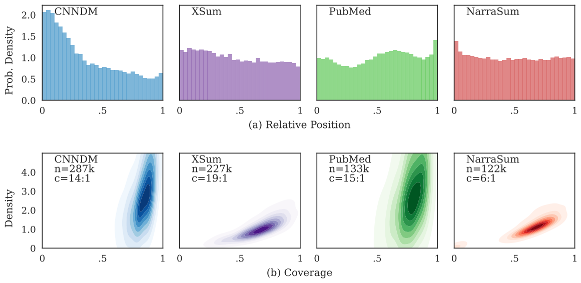


Figure 3: The upper figures show the relative positions of bi-grams of the gold summary in the document. The summary content of NARRASUM is more uniformly distributed over the entire document. The lower figures show the Coverage-Density plots. Compared with CNNDM and PubMed, the summary abstractiveness of NARRASUM is more close to XSum.

NARRASUM contains 122K instances from 22.8K unique movies and 28.5K unique TV episodes, which is ten times larger than the previous largest narrative summarization dataset. We provide the distribution of production years and genres of these movies or TV series in Figure 2, which illustrates that NARRASUM spans a wide time period and contains a broad range of genres. The average length of documents and summaries are 785.97 and 147.06 tokens, and the average compression ratio is 5.34. Most of the documents in NARRASUM are longer than 512 tokens, which is the maximum input length of many pre-trained language models. However, the average length of documents in NARRASUM is still shorter than that of a typical novel chapter (∼5K). This requires the models to process long, but not prohibitively long, inputs while exposing them to the challenges of narrative summarization.

3.2 Summary Characteristics

Different from news articles, salient information in a narrative spreads across the entire text. To verify whether NARRASUM's summaries have this property, we first check the **distribution of the salient information** in the documents. Similar to Kim et al. (2019), we use bi-grams of summary text to represent the salient content of the narrative and then obtain their normalized positions in the documents. Figure 3(a) shows the probability density distribution of the positions of the salient information. We compare the distribution of NARRASUM with CNNDM, XSum, and PubMed. Figure 3(a) indicates that while the salient information of CNNDM and PubMed are concentrated at certain parts of the document, the salient information of NARRASUM is more uniformly distributed over the entire document. It supports our claim that the summarization of NARRASUM requires an understanding of the entire document. There is no lead bias in XSum because the first sentence of the document is removed and is regarded as the summary. It further demonstrates that the first sentence of a news document is enough to summarize the entire document. The section-wise bias in scientific papers is discussed by Gidiotis and Tsoumakas (2020).

Next, we measure the **abstractiveness of summaries** in NARRASUM. To this end, we calculate the Coverage and Density of each summary as suggested by Grusky et al. (2018). Lower Coverage and Density scores indicate that the summary is more abstractive. The distribution is shown in Figure 3(b). The comparison shows that the summaries of NARRASUM are more abstractive than CNNDM and PubMed while being similar to XSum, the most abstractive dataset for news summarization.

We also report the percentage of novel n-grams that are included in the summary but not in the document. A higher percentage of novel n-grams implies a more abstractive summary. As shown in Table 2, the percentage of novel n-grams in NARRASUM is higher than CNNDM and PubMed, and is similar to XSum. This is in line with our observation from the Coverage-Density plot (Figure 3(b)). The difference is that XSum is a news summarization dataset with short summaries (one sentence). NARRASUM is a narrative summarization dataset, where the summaries are of varying length.

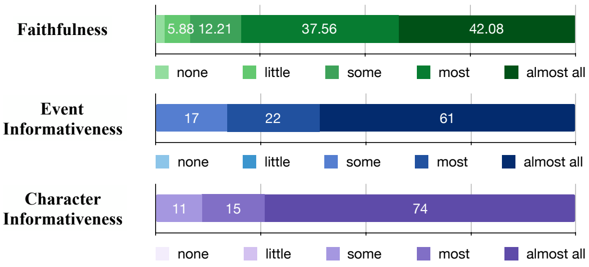


Figure 4: Human assessment results of the quality of NarraSum.

3.3 Quality Assessment

We further conduct a human evaluation to better assess the quality of the NarraSum. We randomly select 100 instances from the test set. For each instance, we ask three workers on Amazon Mechanical Turk to evaluate the summary in terms of *faithfulness* and *informativeness*. For faithfulness, we show annotators each summary sentence and ask them to evaluate how much of the information in this summary sentence is presented in the document. This is a precision-oriented measure and is commonly used for summary evaluation (Lu et al., 2020). For informativeness, we ask annotators to first identify the most salient events and major characters from the document and then evaluate how much of that is covered by the summary. This is a recall-oriented measure. Both human evaluations are collected on a Likert scale of 1-5 (1 means "none", and 5 means "almost all").

To control the annotation quality, we require human judges to be in the United States, and have more than 1,000 HITs approved with an approval rate higher than 98%. We randomly check the annotation results and block the human judges who continually provide low-quality annotations. Human judges were paid a wage rate of $12 per hour, which is higher than the local minimum wage rate.

Figure 4 shows the distributions of human evaluation results. It shows that 80% of content in the summary is faithful to the document. For informativeness, 83% and 89% of summaries cover most of the salient events and characters, respectively. It demonstrates that NarraSum is of high quality in both faithfulness and informativeness, and can foster further research on narrative summarization.

4 Baseline Models

We investigate the performance of several baselines and state-of-the-art neural summarization models on NarraSum. We include both extractive and abstractive models. For extractive models, we use the following methods:

Random selects n sentences from the document randomly.

Lead selects the top-n sentences from the document to compose the summary. This is a strong baseline for news summarization.

TextRank (Mihalcea and Tarau, 2004) is a graph-based extractive summarization model based on PageRank (Brin and Page, 1998) in a graph representation of sentences.

LexRank (Erkan and Radev, 2004) is another graph-based extractive summarization model based on eigenvector centrality .

HSG (Wang et al., 2020) is a heterogeneous graph-based neural extractive summarization model that uses word co-occurrence to enhance sentence contextual representation.

PreSumm (Liu and Lapata, 2019) relies on a pre-trained language model to enhance the sentence representation during text encoding and extractive summarization. We choose **BERT** (Devlin et al., 2019), **RoBERTa** (Liu et al., 2019), and **Longformer** (Beltagy et al., 2020) as the pre-trained models. BERT and RoBERTa limit the input length to be shorter than 512 tokens, while Longformer can accept up to 4,096 tokens.

For abstractive models, we use the following pre-trained sequence-to-sequence models: **BART** (Lewis et al., 2020), **T5** (Raffel et al., 2020), **Pegasus** (Zhang et al., 2020), and **LED** (Beltagy et al., 2020). The input length of the first three models is limited to 512 (base version) or 1,024 (large version). LED uses Longformer as the encoder and therefore can accept up to 4,096 tokens as input.

5 Experiments

5.1 Settings

We conduct experiments with models described in Section 4 to evaluate their performances on NarraSum. For extractive models, we follow the hyper-parameters of the original implementations. For abstractive models, we implement them using the Transformer library (Wolf et al., 2020). We fine-tune each model on the training set of NarraSum with AdamW optimizer (Loshchilov and Hutter, 2019) and batch size of 64. We conduct a simple hyper-parameter search for the learning rate from $\{3e^{-4}, 1e^{-4}, 3e^{-5}\}$ based on the validation loss. We also adopt early stopping based on the val-

Model	R-1	R-2	R-L	SC
Extractive				
Rand	33.94	5.38	29.80	-
Lead	35.11	6.71	30.82	-
LexRank	34.22	5.78	29.70	-
TextRank	34.95	6.18	30.28	-
HSG	36.94	7.54	32.35	-
Bert-Base	36.34	7.29	31.71	-
Roberta-Base	36.47	7.31	31.80	-
LFormer-Base	**37.54***	**7.83***	**32.69***	-
Oracle	42.42	11.44	36.65	-
Abstractive				
Bart-Base	35.81	7.49	31.72	65.19
T5-Base	36.37	7.42	32.17	76.38
LED-Base	37.32	8.14	33.05	62.63
Bart-Large	36.80	8.20	32.62	**77.41***
T5-Large	37.67	8.11	**33.40**	74.14
Pegasus-Large	36.97	7.93	32.64	75.23
LED-Large	**37.71**	**8.87***	33.34	66.91

Table 3: Summarization results evaluated on test set of NarraSum over ROUGE 1 (R-1), ROUGE 2 (R-2), ROUGE L (R-L), and SummaC (SC). SC is only used to evaluate abstractive summaries as extractive summaries are faithful by design. We highlight the best scores separately for extractive and abstrative systems. * indicates a statistically significant difference compared with the second best score (bootstrap resampling, $p < 0.05$ (Koehn and Monz, 2006)).

idation loss to avoid overfitting. During inference, we use beam search with beam-size 5. Our model was trained on a single Quadro RTX 5000 GPU in up to 34 hours, depending on the model size.

Evaluation. We evaluate the generated summaries using ROUGE F_1 score.[8] We further include SummaC (Laban et al., 2022), an automatic measure for summary faithfulness. It achieves state-of-the-art on the benchmark of summary inconsistency detection, and is feasible to be applied to long input and output.

5.2 Automatic Results

Table 3 shows the results on NarraSum using extractive and abstractive summarization approaches.

Extractive Models. The supervised extractive methods outperform the unsupervised extractive methods (the first four models) on all measures by a large margin, indicating that NarraSum can provide a strong supervision signal for identifying the salient information and creating the summary accordingly. PreSumm-BERT or PreSumm-Roberta models underperform HSG because these models have a maximum input length of 512 tokens whereas HSG can accept inputs with arbitrary length. Longformer achieves the best performance on extractive summarization by combining the advantage of pre-training and long document processing. However, there is still a large gap between Longformer's performance and the oracle upper-bound, indicating the challenges in narrative summarization.

Abstractive Models. Among these models, no particular model consistently outperforms others on all subsets. Larger models consistently outperform smaller models, which is inline with previous research. T5 outperforms BART on most Rouge scores, as they adopt summarization-specific pre-training objectives. LED outperforms other models on Rouge due to its ability to encode longer documents. This is consistent with the result of extractive summarization. However, LED performs worst on SummaC-based faithfulness evaluation. This indicates that though the model can process longer documents, understanding and faithfully summarizing lengthy texts is still challenging.

Compression Degree. To better understand the models' capability under different compression degrees, we split the test set into three similar-sized subsets based on the compression ratio of the summary. We then re-evaluate models on each subset separately. We provide details of data split and model performance in Appendix A.1. Results show that it is more challenging to create a short summary than a long one. Other observations on the entire test set still hold across subsets with different levels of compression.

5.3 Human Evaluation

We further conduct a human evaluation on Amazon Mechanical Turk to better understand the models' behaviors and the challenges of this task. We randomly sample 100 instances from the test set and then evaluate the outputs of the best two systems (T5-Large and LED-Large) based on the following four dimensions.

- Fluency: whether or not the summary is grammatically correct and free of repetition;
- Faithfulness: whether or not the summary is faithful to the original document;
- Coherence: whether or not the plot of the narrative summary is logically coherent;

[8] https://github.com/google-research/google-research/tree/master/rouge.

Model	T5-Large	LED-Large
Fluency	4.19	4.11
Faithfulness	3.34	3.23
Coherence	2.87	3.06
Informativeness	2.44	2.67

Table 4: Human evaluation of the generated summaries.

- Informativeness: whether or not the summary reflects the salient events and characters in the original document;

For each instance, we show annotators the original document and the generated summaries. We ask annotators to rate summaries using a 5-point Likert scale and report the average score over all instances. As shown in Table 4, while the pre-trained abstractive models are good at Fluency, they still struggle with other dimensions such as Faithfulness, Coherence, and Informativeness. It further indicates that narrative summarization is a challenging task for current models. In general, the summaries created by T5 are more fluent and faithful, while those created by LED are more coherent and informative. In appendix A.2, we provide examples of generated summaries by various systems.

6 Analysis

We perform a series of analyses about the summary position and character consistency. For a fair comparison among models, we only choose test instances where the length of the document is shorter than the maximum input length of these models (1,024 tokens).

6.1 Analysis of Summary Position

A good narrative summary should preserve the original narrative structure that contains a start, middle, and ending of the narrative. To investigate this, we adopt the method in Kim et al. (2019) to analyze the normalized position of summary bi-grams in the document, where 0 and 1 represent the start and ending of the document, respectively.

Figure 5 shows that while the relative position of n-grams in gold summary is more close to uniformly distributed (Figure 3(a)), the generated summaries are still biased towards the beginning of the original document. It indicates that current models have difficulty understanding the entire documents and preserving the narrative structures.

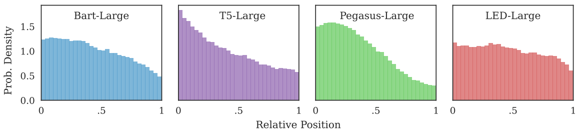


Figure 5: The relative positions of bi-grams of the predicted summaries in the document.

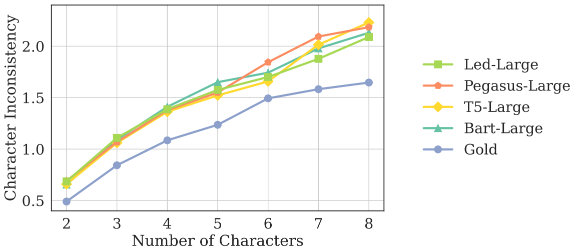


Figure 6: Character inconsistency between documents and summaries w.r.t. the number of characters in the document.

6.2 Character-Wise Analysis

Characters are essential for narratives. Since characters are not considered in Rouge scores, here we propose to measure character consistency by examining whether the major characters in the document are also mentioned in the summary. We assume that major characters appear more frequently in the narrative text. By comparing the distance between the frequency distributions of characters from the document and the summary, we can understand how well the summary includes the major characters of the document.

To this end, we first identify characters from the narrative. We run a coreference resolution model to extract clusters of entity mentions, and we only keep person entities to obtain clusters of characters.[9] We regard each cluster size as the frequency of the corresponding character and then normalize it as a probability. We measure the character inconsistency as the cross-entropy (CE) between the two frequency distributions of characters. A higher CE implies a higher character inconsistency.

In Figure 6, we group the test instances of NarraSum based on the number of distinct characters, and show the cross-entropy of the gold summary and the generated summaries. Compared with the gold summaries, the generated summaries are less consistent with the document at the character level. In general, the difference of cross-entropy between gold summary and generated summaries increases

[9] We use CoreNLP for coreference resolution and named entity recognition.

Evaluated → Trained ↓	MCTest Accuracy	MovieQA Accuracy	LiSCU Accuracy	CBT Accuracy	QuAIL Accuracy	Reddit Rouge-1
NovelChapter	69.66	54.60	25.81	79.90	56.95	28.91
BookSum	70.50	55.21	26.75	80.24	56.33	26.08
NARRASUM	71.83	56.64	26.85	80.66	57.37	32.80

Table 5: Zero-shot performance (Accuracy or Rouge-1) of the model trained on NarraSum and those on other summarization datasets.

Model	R-1	R-2	R-L
Novel Chapter	32.56	6.83	16.25
w/ NARRASUM pretraining	32.88	6.80	16.19
BookSum-Paragraph	21.17	4.35	16.78
w/ NARRASUM pretraining	21.83	4.86	17.13

Table 6: Model performance on Novel Chapter and BookSum-Paragraph with and without pretraining on NARRASUM.

as the number of characters increases, indicating that it is harder for the summarizer to keep the character-level consistency when the document describes more characters.

7 Application to Other Tasks

Besides presenting NARRASUM as a benchmark for narrative summarization, we further explore the broader benefits of this dataset to narrative-related tasks. We first investigate whether pre-training on NARRASUM can improve performance on other narrative summarization tasks. To this end, we first pre-train a BART-Large model on NARRASUM and then finetune it on Novel Chapter and BookSum-Paragraph. We compare with the finetuned models without pre-training on NARRASUM. As shown in Table 6, pre-training on NARRASUM can improve model performance on both datasets, indicating that NARRASUM is beneficial to other narrative summarization tasks.

We then investigate if NARRASUM can help the model learn general knowledge of narrative understanding and summarization. For this, we first pre-train a BART-Large model on NARRASUM and then apply it to several downstream tasks in a zero-shot manner. We choose five tasks that are designed for narrative understanding, i.e., MCTest (Richardson et al., 2013), MovieQA (Tapaswi et al., 2016) , LiSCU (Brahman et al., 2021), CBT (Hill et al., 2016), and QuAIL (Rogers et al., 2020), and one task for narrative summarization, i.e., Reddit TIFU (Kim et al., 2019). For each task, we provide the corresponding task description, method, and evaluation measure in Appendix A.3.

We use models trained on the summarization task to solve these tasks in a zero-shot manner. In other words, we do not use any training data from these tasks. For discriminative tasks, we first convert the (question, answer) pair into a statement using a T5 model (Chen et al., 2021), and then evaluate the probability of generating each statement conditioned on the document (Zhao et al., 2022b). We choose the candidate with the highest generation probability as the predicted answer. Models are evaluated using Accuracy. For the summarization task, we directly apply the trained model to create the summary. Models are evaluated using the Rouge-1 F measure.

We compare the model pre-trained on NARRASUM with those pre-trained on other narrative summarization datasets such as Novel Chapter and BookSum. As shown in Table 5, the model pre-trained on NARRASUM achieves better performance on all narrative-related downstream tasks compared with those pre-trained on other datasets. It indicates that NARRASUM contains high-quality knowledge about narrative understanding and summarization, which can be beneficial to general narrative-related tasks as well.

8 Conclusion

We present NARRASUM, a large-scale narrative summarization dataset that contains plot descriptions of movies and TV episodes and the corresponding summaries. Narratives in NARRASUM are of diverse genres, and the summaries are highly abstractive and of varying lengths. Summarizing the narratives in NARRASUM requires narrative-level understanding, which poses new challenges to current summarization methods. Experiments show that current models struggle with creating high-quality narrative summaries. We hope that NARRASUM will promote future research in text summarization, as well as broader NLP studies such as machine reading comprehension, narrative understanding, and creative writing.

Acknowledgements

This work was supported in part by NSF grant DRL-2112635. We thank anonymous reviewers for their thoughtful and constructive reviews.

Limitations

One limitation of NARRASUM, similar to other automatically constructed datasets, is that we cannot guarantee the entire faithfulness of the summary to the document. To alleviate this issue, we first collect a large-scale dataset and then apply strict rules to select a high-quality subset. The human evaluation and the comparison with other datasets demonstrate that it is worth the trade-off. Another limitation is that NARRASUM does not cover all narrative types such as books, scripts, and personal stories. For those purposes, we suggest readers explore other summarization datasets (Gorinski and Lapata, 2015; Ouyang et al., 2017; Kim et al., 2019; Ladhak et al., 2020; Papalampidi et al., 2020; Kryściński et al., 2021; Chen et al., 2022).

Broader Impact

Besides the contribution to the research field of text summarization, this dataset may spark interest in a broader NLP community. For example, in machine reading comprehension, our paired plot descriptions with low lexical overlap can improve the model's capacity of complex reasoning and understanding (Saha et al., 2018). In narrative understanding, a summary of the narrative can help identify the salient event (Zhang et al., 2021b) as well as the causal, temporal, and hierarchical relationships of events (Hidey and McKeown, 2016; Yao et al., 2020). In creative writing and storytelling, this dataset can support the research of expanding a short story outline to a more detailed story (Ammanabrolu et al., 2020).

We collect and use the publicly available resources for research purposes only, which belong to fair use. This dataset should not be deployed in the real world as anything other than a research prototype, especially commercially.

There is the possibility of (potentially harmful) social biases that can exist in the movies or TV episodes and therefore be introduced in the dataset. While such biases have a limited impact on summarization systems (e.g., introducing harmful biases to the summary when there are no such biases in the document), we suggest the users evaluate the biases and their impacts on their downstream tasks such as creative writing and storytelling, and to make modifications to either the dataset or their models accordingly to avoid such biases.

References

Prithviraj Ammanabrolu, Ethan Tien, Wesley Cheung, Zhaochen Luo, William Ma, Lara J. Martin, and Mark O. Riedl. 2020. Story realization: Expanding plot events into sentences. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7375–7382. AAAI Press.

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

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

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. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1734–1752, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117.

Jifan Chen, Eunsol Choi, and Greg Durrett. 2021. Can NLI models verify QA systems' predictions? In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3841–3854, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mingda Chen, Zewei Chu, Sam Wiseman, and Kevin Gimpel. 2022. SummScreen: A dataset for abstractive screenplay summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8602–8615, Dublin, Ireland. Association for Computational Linguistics.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,* *Volume 2 (Short Papers)*, pages 615–621, New Orleans, Louisiana. Association for Computational Linguistics.

Linguistic Data Consortium and New York Times Company. 2008. *The New York Times Annotated Corpus*. LDC corpora. Linguistic Data Consortium.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479.

Angela Fan, David Grangier, and Michael Auli. 2018. Controllable abstractive summarization. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 45–54, Melbourne, Australia. Association for Computational Linguistics.

Edward Morgan Forster. 1985. *Aspects of the Novel*, volume 19. Houghton Mifflin Harcourt.

Gustav Freytag. 1908. *Freytag's technique of the drama: an exposition of dramatic composition and art*. Scott, Foresman and Company.

Alexios Gidiotis and Grigorios Tsoumakas. 2019. Structured summarization of academic publications. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 636–645. Springer.

Alexios Gidiotis and Grigorios Tsoumakas. 2020. A divide-and-conquer approach to the summarization of long documents. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:3029–3040.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79, Hong Kong, China. Association for Computational Linguistics.

Philip John Gorinski and Mirella Lapata. 2015. Movie script summarization as graph-based scene extraction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1066–1076, Denver, Colorado. Association for Computational Linguistics.

Max Grusky, Mor Naaman, and Yoav Artzi. 2018. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 708–719, New Orleans, Louisiana. Association for Computational Linguistics.

Wynford Hicks, Adams Sally, Harriett Gilbert, Tim Holmes, and Jane Bentley. 2016. *Writing for journalists*. Routledge.

Christopher Hidey and Kathy McKeown. 2016. Identifying causal relations using parallel Wikipedia articles. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1424–1433, Berlin, Germany. Association for Computational Linguistics.

Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. 2016. The goldilocks principle: Reading children's books with explicit memory representations. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Chris Kedzie, Kathleen McKeown, and Hal Daumé III. 2018. Content selection in deep learning models of summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1818–1828, Brussels, Belgium. Association for Computational Linguistics.

Byeongchang Kim, Hyunwoo Kim, and Gunhee Kim. 2019. Abstractive summarization of Reddit posts with multi-level memory networks. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2519–2531, Minneapolis, Minnesota. Association for Computational Linguistics.

Philipp Koehn and Christof Monz. 2006. Manual and automatic evaluation of machine translation between European languages. In *Proceedings on the Workshop on Statistical Machine Translation*, pages 102–121, New York City. Association for Computational Linguistics.

Wojciech Kryściński, Nazneen Rajani, Divyansh Agarwal, Caiming Xiong, and Dragomir Radev. 2021. Booksum: A collection of datasets for long-form narrative summarization. *ArXiv preprint*, abs/2105.08209.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. 2022. SummaC: Re-visiting NLI-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177.

Faisal Ladhak, Bryan Li, Yaser Al-Onaizan, and Kathleen McKeown. 2020. Exploring content selection in summarization of novel chapters. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5043–5054, Online. Association for Computational Linguistics.

Wendy G Lehnert. 1981. Plot units and narrative summarization. *Cognitive science*, 5(4):293–331.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

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.

Deborah L Linebarger and Jessica Taylor Piotrowski. 2009. Tv as storyteller: How exposure to television narratives impacts at-risk preschoolers' story knowledge and narrative skills. *British journal of developmental psychology*, 27(1):47–69.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. 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 3730–3740, Hong Kong, China. 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*, abs/1907.11692.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Yao Lu, Yue Dong, and Laurent Charlin. 2020. Multi-XScience: A large-scale dataset for extreme multi-document summarization of scientific articles. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8068–8074, Online. Association for Computational Linguistics.

Inderjeet Mani. 2012. Computational modeling of narrative. *Synthesis Lectures on Human Language Technologies*, 5(3):1–142.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018a. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018b. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Jessica Ouyang, Serina Chang, and Kathy McKeown. 2017. Crowd-sourced iterative annotation for narrative summarization corpora. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 46–51, Valencia, Spain. Association for Computational Linguistics.

Pinelopi Papalampidi, Frank Keller, Lea Frermann, and Mirella Lapata. 2020. Screenplay summarization using latent narrative structure. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1920–1933, Online. Association for Computational Linguistics.

Pinelopi Papalampidi, Frank Keller, and Mirella Lapata. 2019. Movie plot analysis via turning point identification. 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 1707–1717, Hong Kong, China. Association for Computational Linguistics.

Andrew Piper, Richard Jean So, and David Bamman. 2021. Narrative theory for computational narrative understanding. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 298–311, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Gerald Prince. 1973. *A Grammar of Stories: An Introduction.*

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. *Journal of Machine Learning Research*, 21(140):1–67.

Matthew Richardson, Christopher J.C. Burges, and Erin Renshaw. 2013. MCTest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 193–203, Seattle, Washington, USA. Association for Computational Linguistics.

Anna Rogers, Olga Kovaleva, Matthew Downey, and Anna Rumshisky. 2020. Getting closer to AI complete question answering: A set of prerequisite real tasks. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8722–8731. AAAI Press.

Amrita Saha, Rahul Aralikatte, Mitesh M. Khapra, and Karthik Sankaranarayanan. 2018. DuoRC: Towards complex language understanding with paraphrased reading comprehension. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1683–1693, Melbourne, Australia. Association for Computational Linguistics.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Wei Shen, Jianyong Wang, and Jiawei Han. 2014. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering*, 27(2):443–460.

Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2016. Movieqa: Understanding stories in movies through question-answering. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 4631–4640. IEEE Computer Society.

Danqing Wang, Pengfei Liu, Yining Zheng, Xipeng Qiu, and Xuanjing Huang. 2020. Heterogeneous graph neural networks for extractive document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6209–6219, Online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yu Xiong, Qingqiu Huang, Lingfeng Guo, Hang Zhou, Bolei Zhou, and Dahua Lin. 2019. A graph-based framework to bridge movies and synopses. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 4591–4600. IEEE.

Wenlin Yao, Zeyu Dai, Maitreyi Ramaswamy, Bonan Min, and Ruihong Huang. 2020. Weakly Supervised Subevent Knowledge Acquisition. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5345–5356, Online. Association for Computational Linguistics.

Wenpeng Yin, Dragomir Radev, and Caiming Xiong. 2021. DocNLI: A large-scale dataset for document-level natural language inference. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4913–4922, Online. Association for Computational Linguistics.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339. PMLR.

Shiyue Zhang, Asli Celikyilmaz, Jianfeng Gao, and Mohit Bansal. 2021a. EmailSum: Abstractive email thread summarization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6895–6909, Online. Association for Computational Linguistics.

Xiyang Zhang, Muhao Chen, and Jonathan May. 2021b. Salience-aware event chain modeling for narrative understanding. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1418–1428, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Chao Zhao, Tenghao Huang, Somnath Basu Roy Chowdhury, Muthu Kumar Chandrasekaran, Kathleen McKeown, and Snigdha Chaturvedi. 2022a. Read top news first: A document reordering approach for multi-document news summarization. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 613–621, Dublin, Ireland. Association for Computational Linguistics.

Chao Zhao, Wenlin Yao, Dian Yu, Kaiqiang Song, Dong Yu, and Jianshu Chen. 2022b. Learning-by-narrating: Narrative pre-training for zero-shot dialogue comprehension. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 212–218, Dublin, Ireland. Association for Computational Linguistics.

Subsets	Size	L-doc	L-sum	Factor	Ratio
Validation					
Low Comp.	1,837	461	170	0.37	2.70
Medium Comp.	2,161	704	152	0.22	4.55
High Comp.	1,736	1,290	103	0.09	11.11
Test					
Low Comp.	1,773	443	164	0.37	2.70
Medium Comp.	2,160	696	151	0.22	4.55
High Comp.	1,590	1,355	108	0.09	11.11

Table 7: Statistics of validation and test subsets. The compression factor (Factor) is defined as the length ratio between the summary to the document. The compression ratio (Ratio) is defined as the length ratio between the document to the summary.

Ming Zhong, Pengfei Liu, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2019. Searching for effective neural extractive summarization: What works and what's next. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1049–1058, Florence, Italy. Association for Computational Linguistics.

Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir Radev. 2021. QMSum: A new benchmark for query-based multi-domain meeting summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5905–5921, Online. Association for Computational Linguistics.

A Appendix

A.1 Compression-aware Evaluation

Intuitively, creating a short summary is more challenging than creating a long one, as it requires selecting information more precisely and textualizing the salient information more abstractively (for abstractive models only).

To better understand the models' capability under different degrees of compression, we split the validation set and the test set into three subsets based on the compression factor, α, of the gold summary. The compression factor is defined as the length ratio between the summary to document. Specifically, we regard $\alpha < 0.15$ as **high compression**, $0.15 \leq \alpha < 0.3$ as **medium compression**, and $\alpha \geq 0.3$ as **low compression**. Using these threshold values, we can split the validation and test sets into three similar-sized subsets. We list the detailed statistics of each subset in Table 7.

During inference, the desired length of the summary for a given document is determined by multiplying the document's length (the number of tokens in the document) with the α for the desired compression level. The αs for the three compression levels are determined using their average values in the corresponding validation subsets (0.37, 0.22, and 0.09 for the low, medium, and high compression, respectively). For extractive models, we continually add sentences to the summary according to their predicted salience until the summary length is most close to the desired length (either longer or shorter). For the abstractive models, we roughly control the length of the summary using the method described by Fan et al. (2018). The basic idea is to split the entire dataset into n equally-sized bins according to the summary length. For each data instance, the ID of the corresponding bin is appended to the front of the document to indicate the desired scope of summary length. We use $n = 10$.

Table 8 shows the results on NarraSum using extractive and abstractive summarization approaches for different degrees of compression. Generally, with the increase in the compression ratio, the Rouge scores become lower. It indicates that it is more challenging to create a short summary compared with a long one. However, observations on the entire test set still hold across different levels of compress degree.

A.2 Qualitative Analysis

Table 9 shows an example with the narrative document, gold summary, and the predicted summaries. The narrative document is from Season 2, Episode 1 of *Zoey 101*, an American comedy-drama TV.

This example shows that while the gold summary can faithfully cover the most salient information from the narrative document, summaries generated by machines contain some errors. Bart does not contain the information of "*Zoey returns to PAC*" and "*Dana will not return*". T5 fails to follow the causal and temporal relationships of events. The summary created by Pegasus is generally not coherent. The summary created by LED covers all important information but the writing is not fluent.

A.3 Zero-Shot Tasks

We choose five tasks that are designed for narrative understanding (MCTest, MovieQA, LiSCU, CBT, and QuAIL), and one task for narrative summarization (Reddit TIFU). We don't include tasks that are not related to narrative. In this section, we describe the details of these datasets.

MCTest (Richardson et al., 2013) is a dataset designed for open-domain multiple-choice reading

Model	Low Comp (Long Summ.)				Medium Comp. (Medium Summ.)				High Comp. (Short Summ.)			
	R-1	**R-2**	**R-L**	**SC**	**R-1**	**R-2**	**R-L**	**SC**	**R-1**	**R-2**	**R-L**	**SC**
Extractive												
RAND	37.19	7.14	32.64	-	34.79	5.35	30.72	-	29.16	3.43	25.38	-
LEAD	38.62	8.45	33.88	-	36.18	6.87	31.92	-	29.73	4.54	25.9	-
LEXRANK	36.90	7.14	32.11	-	35.11	5.82	30.68	-	30.02	4.17	25.66	-
TEXTRANK	37.55	7.67	32.77	-	35.89	6.23	31.22	-	30.77	4.43	26.21	-
HSG	39.86	9.06	34.97	-	38.00	7.74	33.47	-	32.25	5.54	27.90	-
BERT-B	39.00	8.82	34.04	-	37.51	7.50	32.91	-	31.77	5.29	27.49	-
ROBERTA-B	39.08	8.78	34.06	-	37.60	7.50	32.96	-	32.01	5.40	27.69	-
LFORMER-B	**40.38***	**9.41***	**35.26**	-	**38.69***	**8.08***	**33.88***	-	**32.99***	**5.85***	**28.27***	-
ORACLE	43.55	12.48	37.89	-	43.36	11.6	37.72	-	39.88	10.06	33.81	-
Abstractive												
BART-B	38.21	8.80	33.8	71.94	36.62	7.51	32.60	66.85	32.02	5.99	28.21	57.14
T5-B	38.56	8.74	34.14	81.43	37.34	7.60	33.23	78.07	32.61	5.68	28.53	**69.81***
LED-B	39.46	9.22	34.93	74.01	38.10	8.18	33.96	66.85	33.58	6.83	29.53	47.34
BART-L	39.29	9.37	34.87	**86.86***	36.92	8.04	32.94	**81.48***	33.84	7.10	29.67	64.05
T5-L	39.39	9.17	34.94	84.09	**38.49***	8.22	**34.29***	75.65	34.65	6.79	30.46	63.43
PEGASUS-L	39.21	9.14	34.67	84.70	37.57	7.86	33.34	80.70	33.65	6.67	29.40	60.13
LED-L	**39.57**	**9.74***	**35.06**	78.78	37.86	**8.73***	33.65	71.43	**35.41***	**8.11***	**30.99***	50.83

Table 8: Summarization results evaluated on three test subsets of NARRASUM over ROUGE 1 (R-1), ROUGE 2 (R-2), ROUGE L (R-L), and SummaC (SC). We highlight the best scores separately for extractive and abstrative systems. * indicates a statistically significant difference compared with the second best score (bootstrap resampling, $p < 0.05$ (Koehn and Monz, 2006)).

comprehension. The dataset contains 500 fictional stories, with four multiple choice questions per story.

CBT (Hill et al., 2016) is also an dataset designed for open-domain reading comprehension. The dataset builds question-answer pairs from 108 children's books with clear narrative structure.

MovieQA (Tapaswi et al., 2016) aims to evaluate models' ability of automatic story comprehension. The dataset consists of 14,944 multiple-choice questions sourced from 408 movies. Each question has five options. We use the movie summaries as input to answer these questions.

LiSCU (Brahman et al., 2021) is a character-centric narrative understanding task to test the model performance from the perspective of characters. This dataset contains 1,708 literature summaries and 9,499 character descriptions. Given the literature summary, the model needs to identify the character's name from an anonymized character description and a list of character candidates.

QuAIL (Rogers et al., 2020) is a machine reading comprehension benchmark with varying types of reasoning. Solving this challenge requires an understanding of not only the text-based information from the document but also the world knowledge and commonsense knowledge. Documents in QuAIL are collected from fiction, user stories, and so on. Each question has four options.

Reddit TIFU (Kim et al., 2019) is an abstractive summarization dataset. It consists of 120K crowd-generated posts from the online discussion forum Reddit, as well as their corresponding summaries. Different from other narrative summarization datasets we discussed in the paper, narratives in Reddit TIFU are mostly written in informal and conversational text, and the story is about the poster doing something wrong or messing everything up. These features make Reddit TIFU a good out-of-domain test data to evaluate the models' generalization power for narrative summarization.

Document: It is the girls' second year at PCA. Dana has been accepted to a European Exchange Program, so Zoey and Nicole think will have their dorm room to themselves. However, Coco informs them they will be getting a new roommate. Zoey and Nicole go to the housing office hoping to choose a roommate, but their request is denied (After Nicole accidentally destroy's the secretary's perfume collection). When they return to their dorm, they find their new roommate, Lola Martinez, a goth-punk girl who disturbs them by drinking raw eggs and claiming to talk to the dead. As Lola's behavior grows more erratic, Zoey and Nicole are out of the dorm, but they later discover that Lola is not a goth-punk, but an aspiring actress who managed to fool them with her acting skills. Meanwhile, Michael and Chase have to deal with Logan, whose father has given him a huge entertainment center for their dorm room—which turns out to be a huge headache.
Gold Summary: Zoey and Nicole return to PCA for the second year. But they soon find out Dana will not be returning, and a very strange roommate, Lola, has come to take her place. The year gets to a bad start for Chase and Michael, because of Logan's very loud gifts from his dad.
Bart-large: **[Missing *Zoey returns to PAC* and *DANA will not return*]** Zoey and Nicole get a new roommate, Lola Martinez, a goth-punk girl who disturbs them by drinking raw eggs and claiming to talk to the dead. Meanwhile, Michael and Chase have to deal with Logan, whose father has given him a huge entertainment center for their dorm room–which turns out to be a huge headache.
T5-Large: Zoey and Nicole get a new roommate, Lola Martinez, who turns out to be a goth-punk. Meanwhile, Michael and Chase have to deal with Logan, whose father has given him a huge entertainment center for their dorm room. **[Fail to follow the causal and temporal relationship]** Meanwhile, Dana is accepted to a European exchange program, so the girls think they'll have their own room, but Coco tells them that they'll be getting a roommate.
Pegasus-Large: Zoey and Nicole's first year at PCA is coming to an end, and they're looking forward to living in their own dorm room. However, Coco informs them that Dana has been accepted to a European exchange program, so Zoey and Nicole think **[Not logically coherent]** they will have their own room to themselves. Zoey and Nicole go to the housing office hoping to choose a roommate, but their request is denied (After Nicole accidentally destroy's the secretary's perfume collection).
LED-Large: Zoey and Nicole are thrilled when Dana gets accepted to a European exchange program at PCA, **[Not fluent]** but when they find out they have to share a dorm room with Lola Martinez, a goth-punk girl who drinks raw eggs and talks to the dead. Meanwhile, Michael and Chase have to deal with Logan, whose father has given him a huge entertainment center for their dorm room, which turns out to be a huge headache.

Table 9: Sample summaries generated by baseline models. We show the original document, the gold summary, and the output summaries from four large models. We highlight the typical errors of each output summary.