Universiteit Leiden

Master Computer Science

Opinionated content summarization with Transformer architectures

Name:	Eirini Kousathana
Student ID:	29029591
Date:	July 4, 2023
Specialisation:	Artificial Intelligence
1st supervisor:	Suzan Verberne
2nd supervisor:	Gijs Wijnholds

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

Discussions are more prevalent than ever on online spaces. Automatic summarization can be a tool to provide valuable insights on public opinion and accelerate decision making. Recent advances in abstractive summarization research that leverage pre-trained large language models led to summarizers that generate fluent, human-like abstracts. However, online discussions are an under-investigated domain, mainly due to the lack of big training datasets with human curated summaries. In this research we are investigating the performance of some state-of-the-art summarization models on the online discussion domain and specifically on opinionated content. We focus on two architecture paradigms: *Multi–document* summarization models and *Long–input* summarization models. For all the models we are using pre-trained checkpoints on variants of masked language modelling tasks. We fine-tune these models on a dataset curated with comments from the New York Times comment section and their respective human curated summaries as ground truth. Subsequently, we evaluate the fine-tuned models on the test subset of the NYT dataset and another dataset containing comments from The Guardian. Our evaluation is quantitative and we pick a small sample of examples that we examine closer in a qualitative manner. Eventually, we observed that Long-input summarization techniques perform better, however the frequency of highly extractive input, hallucinated output and low rate of combining opinions and views points showcased from all the models, leads us to believe that tailored datasets and methods could be explored in the future.

Keywords: Abstractive summarization, Online discussion summarization, Multi-document summarization, Long-input summarization

1 Introduction

Due to the vast amount of daily textual content that is generated and made available online, extracting salient information can facilitate the digestion of this information for an end user. Automatic text summarization is a task that addresses this need, by distilling a piece of text into a shorter form. Automated summarization research has achieved great success [32] [71] on single source summarization benchmarks such as the news summarization dataset like CNN/Dailymail [24] or XSum [43]. This is because news articles are documents that come with a specific structure, such as titles, subtitles and topic sentences that facilitate the construction of datasets with reference summaries on which models can be trained.

With the prevalence of social media and discussion forums, the summarization of opiniated content could reflect in a short, comprehensive way the public opinion, which could accelerate and enhance decision making. Therefore, investigating how recent summarization systems perform on this domain, could provide useful insights for future research.

Extractive approaches, which in the context of online discussion summarization, usually entails that the summary is a compilation of salient posts or sub-parts of the posts (e.g. sentences), have been explored in the literature [70] [59] [63]. Abstractive summarization though - the task of generating novel text that encapsulates the important information of a source text - can generate more coherent summaries that are best received and understood by humans [15].

Despite the recent progress in automatic abstractive summarization with transformer-based architectures, the domain of online conversations has been under-investigated mainly due to the lack of extensive datasets. The state-of-the-art architectures have hundreds of millions or even billion of parameters so they need large training corpora to be able to learn general linguistic attributes and not over–fit the training dataset [18]. There have been successful attempts at curating big automatic social media post summarization datasets such as Reddit posts [55], however these posts are single-authored posts that do not reflect the structure of online discourse. Automated construction of online discussions summaries have been implemented by [69], [58], [57], by curating pseudo-summaries - i.e. automatically generated summary-like text which can not be perceived as a gold standard summary - using meta-information, such as the most liked comment or the title of the post. These summaries can be biased to the opinion expressed in the most popular comment or disregard information that is not related to the original post but is important to the discussion itself. Human curated summaries of online conversations are naturally more reliable but simultaneously more resource-intensive. As a result, this type of datasets [19], [3], consist of few examples. The use of pre-trained Language Models, which leverage the power of transfer learning, that has shown impressive results for the summarization task already [32] [71], seems as a good fit in a domain with limited training examples.

We can define the forum discussion task as a *multi-document* summarization task consisting of multiple short texts that concern a relevant post [50], or as a *long-input* document summarization task [75]. The multi-document approach attempts to capture the cross-document relationships to guide the summarization result. The long-input modelling intends to efficiently process the whole source document. Both modelling techniques have acquired the attention of the research community since the introduction of extensive datasets as the Multi-News dataset [18] for the multi-document summarization task and GovReport [27] for the long-input summarization task. In this report we focus on techniques derived from the above modelling approaches and how they perform in the domain of online discussions.

In this report we aim to answer the following questions:

1. How do recent pre-trained multi-document and long-input summarization techniques perform on the online conversation domain?

2. Can we pinpoint qualitative differences between them?

To answer these questions we plan on fine-tuning state-of-the-art pre-trained summarization models that are designed for multi–document input and for long–input documents. The multi–document techniques include: the `PRIMERA` [67] model, that is pre-trained on a task tailored to multi–document summarization, and the `BARTlong-graph` model that takes auxiliary input extracted from a semantic graph, built on the cluster of documents to be summarized, to capture cross document relationships. The long–input techniques include the `BARTlong` model, that is an extension of the `BART` model that uses sparse self-attention so that it can receive longer input, and the `SUMM-N` model that breaks the summarization process to multiple stages in order to handle longer input. To evaluate the performance of these models, we will quantitatively assess them by measuring their resemblance to the golden truth summaries, as well as by using measures that can provide us insights on qualities like redundancy, consistency, abstractness, ability to summarize dispersed information

within the cluster and the degree of similarity between them. Lastly, we will evaluate them qualitatively by selecting a random sample of them and examining them in greater detail.

2 Background

2.1 The Summarization task

Automatic summarization is the NLP task of condensing a document or a collection of documents into a shorter form that contains the most salient pieces of information. Depending on the nature and size of the input text as well as the desired form of the output summary, the summarization task can take many forms, requiring different approaches.

2.1.1 Extractive vs Abstractive

Extractive summarization refers to the task of compiling a summary by concatenating unaltered units of text of the original source. The units are usually sentences, and an extractive system typically consist of sub-systems of *ranking* and *selecting* sentences for inclusion to the final summary based on their saliency. Determining which sentences are suitable for the output, can be modeled as a sentence classification problem that can be tackled in a supervised [6] or unsupervised [70] fashion. Engineering representations of the sentences to be provided as input to these models is a usual step, either using handcrafted features [1] or neural derived representations [41].

Abstractive summarization, on the other hand, refers to the task of compressing the input text in a transformative way, where ideally the most prominent semantic topics are guiding the generation of novel summary text. As abstractive summarization implies systems that can perform intricate tasks such as paraphrasing and generalisation, semantic analysis structures of the text, such as dependency trees, named entity graphs, ontologies etc, are often part of these systems. They can serve as an intermediate representation of text [9] or auxiliary information [54], to aid summary generation.

Hybrid summarization is the combination of extractive and abstractive summarization, which aims to benefit from the strong features of both of the approaches. A hybrid system usually uses an extractive submodule to select salient content from the source text, content that is then fed to an abstractive sub-module that generates a summary. [54] [31]

It is evident that the extractive approach is a simpler and faster approach than the abstractive one. Moreover, since the text included in an extract is a concatenation of unchanged parts of the input, the information included is faithful to the original, whereas in the case of an abstract, fusion or paraphrasing of sentences can lead to false content [40]. However, the extractive output lacks coherence and the order of the concatenated sentences can lead to confusion of facts, due to phenomena like "dangling" anaphora [23], where it is not evident what pronouns are referring to. Automated abstractive summaries in theory are the closest to a human-generated summary, as they are in principle coherent, fluent pieces of text. Their ability to combine information from the whole source text, leads to greater compression rates compared to extractive approaches [43]. However, generating an abstract is a complicated task, as it does not only require the generation of natural language - a difficult task in itself - but the summarizing system should hold a deep understanding of the text, so that the generated output stays faithful to the source [40].

2.1.2 Single vs Multi-Document

Other than summarizing one source of text – the task defined as *Single Document Summarization* (SDS) – there are a lot of applications of summarizing a collection of text sources that handle the same topic and this task is called *Multi-Document Sumarization* (MDS). Some examples of the MDS task are multiple news articles covering the same issue [18], product reviews [21], Wikipedia entries [37] and more.

The MDS task can be transformed in a SDS task, by concatenating the different documents sequentially to a single document, which afterwards can act as input to a SDS system [18]. This strategy however, does not take into account some particularities of the MDS task. Since we are dealing with many documents, there can be redundant or contradicting information among them. Moreover, a flat concatenation leads to lengthy input, which is problematic for systems whose complexity depends on input length, such as transformer based models [62]. In that sense, summarization systems that use this concatenation strategy, should have the ability to efficiently process long sequences [45]. A hybrid summarization approach, where an extractive

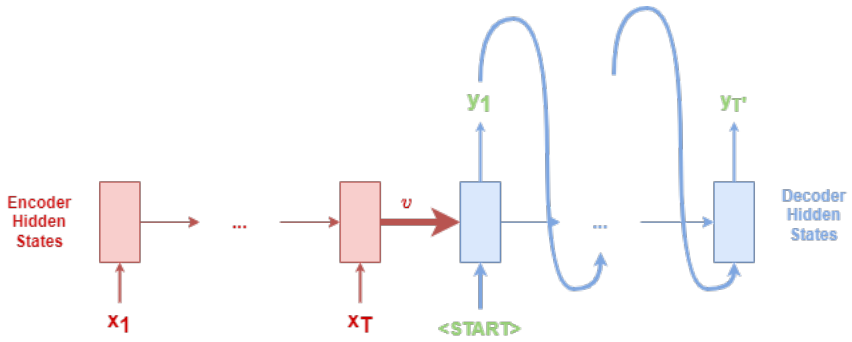


Figure 1: Sequence-to-sequence architecture with a RNN-based encoder and decoder.

system first selects salient information across documents – resulting into a smaller input – which is then fed into an abstractor to form a final summary, is also adopted for the multi-document setting [37].

The challenge that MDS systems have to tackle is to understand and successively exploit the cross-document relationships to generate relevant, non-redundant and coherent summaries. Text has a natural hierarchical structure, where documents consist of paragraphs, paragraphs consist of sentences and sentences consist of words. Some approaches try to capture cross document relationships by producing an intermediate representation of the different text components of the documents and of the relationship between them. Constructing graphs where then nodes represent a text unit and the edges a relationship between them, like cosine similarity [34] or semantic relationships [45], is an intuitive modelling technique [16].

2.2 Neural Abstractive Summarization

Before neural networks became the building blocks of most of the state-of-art systems for NLP tasks [56], abstractive summarization solutions were less explored than their extractive counterparts. Some popular techniques used aimed at transforming sentences, namely *sentence compression* [11] and *sentence fusion* [9], which were based on syntactic and grammar rules. The introduction of the sequence-to-sequence learning paradigm for the machine translation task [56], played a pivotal role for the direction of neural abstractive summarization.

2.2.1 Sequence-to-Sequence Architectures

A sequence-to-sequence neural model, is a model that learns to transform an input sequence to an output sequence. In the case of abstractive summarization, the input sequence is a piece of text and the desired output is a shorter version of that text. The model follows a neural architecture called the *encoder decoder* architecture. The encoder takes the input sequence and produces an intermediate neural representation. The decoder acts like a neural *language model* conditioned on the encoder's output. A Language Model (LM) is a model that learns to predict the next word given a sequence of words. Concretely, given a sequence of words $x^{(1)}, x^{(2)}, ..., x^{(t)}$, compute the probability distribution of the next word $x^{(t+1)}$:

$$P(x^{(t+1)}|x^{(t)}, ..., x^{(1)})$$

where $x^{(t+1)}$ can be any word in the vocabulary $V = w_1, ..., w_{|V|}$ [5]. Therefore, in the case of the decoder, it learns to estimate the conditional probability $p(y_1, ..., y_{T'}|x_1, ..., x_T)$ where $(x_1, ..., x_T)$ is an input sequence and $(y_1, ..., y_{T'})$ is its corresponding output sequence (T and T' may be different lengths). Let v be the representation of the input sequence $(x_1, ..., x_T)$ derived from the encoder, then the probability of the output sequence $(y_1, ..., y_{T'})$ is defined as [56]:

$$p(y_1, ..., y_{T'}|x_1, ..., x_T) = \prod_{t=1}^{T'} p(y_t|v, y_1, ..., y_{t-1})$$

After the application of seq2seq learning to the machine translation task, researchers investigated its application to the summarization task [51]. The implementations that were developed experimented with the internal architecture of the encoder and decoder modules, as well as a variety of mechanisms that tailored the seq2seq paradigm to the summarization task. The neural building blocks that dominated the seq2seq research were

Recurrent neural networks (RNNs). Recurrent neural networks, were designed to process sequences, where each token (e.g. word, sub-word, letter) of the sequence is treated as a computational timestep, meaning that we have as many timesteps as the sequence length in tokens. During a timestep t, a *hidden state* h_t is produced using the t_{th} input token and the $t-1$ hidden state, as well as an output vector o_t utilizing the h_t state vector. In this way RNNs can pass along historic information to future computations. An illustration of a simple RNN based encoder decoder architecture can be seen in Figure 1. Their variants LSTM (Long short-term memory) [25] and Gated Recurrent Unit (GRU) [10] networks, address the issues of inefficient gradient flow of the original architecture, which influences the network's ability to capture long term dependencies.

Aggregating all information about the input sequence into a singular fixed-length vector, can cause an information bottleneck. As a remedy to this issue, the *attention* or else *alignment* mechanism was introduced [2]. The main idea is that direct connections from the decoder to the encoder's hidden states, will allow the most relevant parts of the input to influence the prediction on each decoder step. An outline of how this mechanism can operate follows: given the hidden state of the decoder $\mathbf{s_t}$ in timestep t and the encoder hidden states $\mathbf{h_1}, ..., \mathbf{h_N}$ where N is the input sequence length, we compute *attention scores* by calculating the dot product between each encoder hidden state and the decoder hidden state $\mathbf{s_t}$, which we can represent with a single vector $\mathbf{e^t}$. Then these scores can be turned into a probability distribution over the encoder hidden states, using the `softmax` equation. Subsequently, with the attention distribution, we can compute a weighted sum of the encoder hidden state vectors leading to a single vector, the *context* vector. Lastly, the output of the decoder in timestep t is based on the concatenation of the context vector and $\mathbf{s_t}$. The attention mechanism has become a staple mechanism to include in neural implementations, as it not only solves the bottleneck issue, but it helps with gradient flow through its direct connections as well as the interpretability of deep neural networks [60].

Sub-word tokenization At the early steps of sequence-to-sequence learning research, the vocabulary of the models was a finite set of words, sometimes based on the most frequent words of a language or from a training corpus. During training, every word that was not part of the vocabulary was represented by a special *out-of-vocabulary* token (e.g. `'UNK'`). Accordingly, during inference, unseen words were also mapped that way and so it was also probable that it could be generated as well to the output sequence.

Since natural language can contain different grammar forms, variations and misspellings, assuming a finite vocabulary of words proved to be limiting. For this reason **sub-word** modelling methods were developed, where the sequence is broken down to parts of lower level than words (i.e., part of words, characters, bytes). *Byte pair encoding (BPE)* [53] and *Wordpiece* [66] are favoured methods for defining a sub-word vocabulary of desired length based on a training corpus. In a sub-word vocabulary common words remain intact but rarer words are split into smaller parts, with the worst case scenario being split up into individual characters.

2.2.2 Transformers

Even though RNN-based neural architectures with the attention mechanism dominated the research of sequence-to-sequence tasks, some performance issues stemming from the sequential nature of them, led researchers to explore different neural architectures. Specifically, sequential processing means that two words interact with each other in $\mathcal{O}$(sequence length) time, which practically led to difficulty in learning long-term dependencies [42]. Moreover, sequential calculations are unparallelizable, since future calculations are dependent on previous ones, causing an increased need of time and memory resources.

Vanswani et al. introduced the *Transformer* architecture motivated to tackle the above issues. Transformers are neural network models, that follow an encoder-decoder architecture with a self attention mechanism [62]. An overview of the architecture can be seen in Figure 2. Inside the gray area, the inner architecture of one encoder layer (left) and one decoder layer (right) is depicted. The $N\times$ symbol beside them signifies that the Transformer encoder and decoder can consist of multiple N layers of the same form.

Self Attention To capture the dependencies within a sequence either in the encoder or the decoder, instead of recurrence, the self attention mechanism is employed. Contrary to the encoder-decoder attention we have discussed before, self attention is applied within the encoder or decoder. Attention can generally be described as follows: given a token t to be predicted – represented as a vector – we want to learn how important the tokens within *context* are to this prediction. The t token can be mapped to a **query** vector. Then each context token has a **key-value** pair which are also represented as vectors. Each key vector is scored by how relatable they are to the query vector. If we sum the value vectors by weighting them according to the query-key scores, the result is a new vector representation of the t token that incorporates context information. In the case

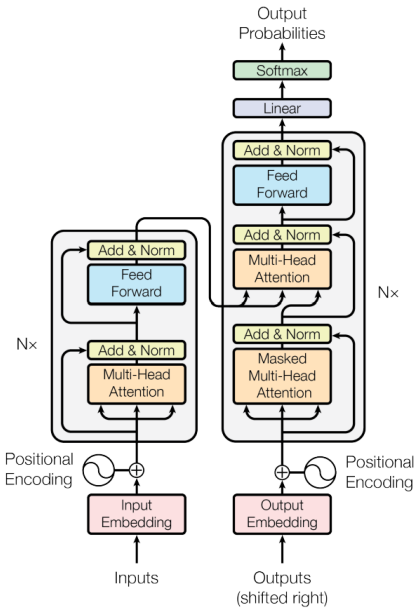


Figure 2: From Vaswani et al. [62]: Architecture of the transformer model.

of self-attention each token of a sequence is considered as query and their context is every other token in the sequence, including itself. The query, key and value vectors are computed by a linear transformation with learnable parameters. The query-key scores are computed with the dot-product operation scaled by d_k, the dimension of the query and key vectors. Since the attention function can be applied independently for each token, the computations can be performed in parallel and efficiently by stacking the query, value, key vectors to form matrices and perform matrix multiplications between them.

Positional Encoding The self attention mechanism inherently does not have any information about how words are ordered in a sequence. For a lot of human languages, word order is important to the meaning of a text. To address this issue, positional encodings are introduced. A sequence of length T can have $[1, ..., T]$ word positions. Each position can be mapped to a vector which can be added to the embedding of a token and inject in this way order information. The positional encodings can be fixed vectors or learnable parameters. Vaswani et al. choose fixed sinusoidal representations where the vectors consisted of sinusoidal functions of varying periods. Learnable encodings, i.e. vector representations that are updated during the training process, have the advantage that they learn to fit the data, rendering them more flexible.

Encoder In Figure 2, the attention sub-module present in the encoder – colored orange – is named *Multi-Head* attention. The self-attention mechanism is included in this sub-module, where the attention function is applied multiple times on sub-vectors of the query, key, value vectors, with separate learnable parameters for each *attention head*. The final output is a linear transformation of the concatenation of the output of the different heads.

Because the self attention mechanism does not contain any non-linear operations, it is succeeded by a fully connected feed forward network with non-linear activation function. The residual connections and layer normalization indicated by the *Add & Norm* sub-module are introduced as they improve training efficiency of deep neural architectures [33].

Decoder In the decoder there is also the *Masked Multi-Head* attention. This also refers to a self attention mechanism. Since the decoder needs to predict tokens autoregressively, i.e. based on the previous values, it shouldn't have access to future tokens. To combat this, attention scores that correspond to illegal query-key pairings are set to $-\infty$. Next, the Multi-Head attention that follows refers to *cross* encoder decoder attention, where the query vectors are drawn from the decoder and the key and value vectors are drawn from the encoder.

2.2.3 Pre-training

Pre-training is a paradigm that significantly propelled the performance of neural NLP models. The main idea is initializing parts or the whole NLP model with weights that are learned through training on a general or relevant task instead of initializing with random weights. Then we can train the model using training data pertaining to the task at hand. The latter step is referred to as *fine-tuning* on a *downstream* task. Leveraging knowledge acquired from one task (pre-training) for another task (fine-tuning) is the machine learning principle called **transfer learning**.

Pre-training methods train the models on the task of reconstructing part of the input using very large bodies of text. Pre-training tasks defer for models that need to make predictions *autoregressively* (i.e., it predicts future values based on past values) and models that makes predictions based on the a complete piece of text. For decoder architectures the pre-training objective of language modelling is a natural fit [48]. They can be used for downstream tasks as generators of one word at a time text given a prompt (e.g. for the summarization task the prompt/context can be the document). For encoder architectures, traditional language modelling is not suitable as the they get bi-directional context. Devlin et at [13] introduced the *Masked Language Modelling* task, where a percentage of tokens from the input are replaced with a special `[MASK]` token or by another random token and subsequently the model is trained to predict those tokens. For encoder-decoder architectures it has been found by Raffel et al. [49] that the pre-training task of masked language modelling leads to better performance for a variety of downstream tasks compared to unidirectional language modelling (i.e., the task of predicting the next word given a sequence of words). Specifically, Raffel et al. used an objective called *span corruption* where random tokens from the input are randomly chosen and every consecutive span of these tokens is replaced with unique special tokens. Then the model is trained to predict these corrupted spans.

When it comes to fine-tuning, training every parameter of a pre-trained model usually yields better performance but can prove to be computationally expensive and can also result in catastrophic forgetting (i.e., the phenomenon where a model forgets the previously learned knowledge [8]). Because of this, more lightweight fine-tuning can take place where a subset of the training parameters is freezed and the rest are trained on the new training data [14] [26].

2.3 Models

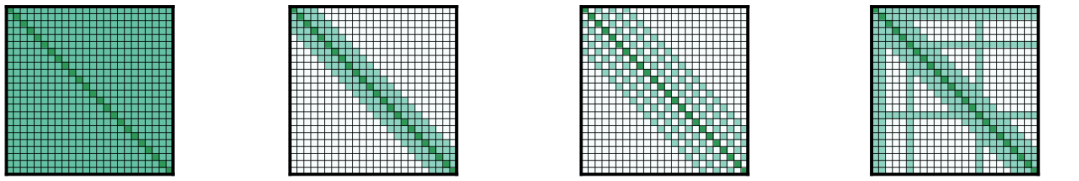
2.3.1 BART

BART stands for Bidirectional and Auto-Regressive Transformer [32], as it consists of a bidirectional encoder and an auto-regressive decoder, which makes this model applicable to many downstream NLP tasks, from sequence classification to summarization. BART's architecture is identical to the original Transformer (see section 2.2.2), with minor differences such as using GeLUs activation functions and initializing parameters from the normal distribution of mean 0 and standard deviation 0.02. The model is trained using a self supervision scheme, where transformations to the input text are performed and the model learns to reconstruct it auto-regressively. The transformations that the authors used to introduce noise to the input documents are Token Masking as seen in BERT [13], Token Deletion, Text Infilling, Sentence Permutation and Document Rotation. The motivation of using a selection of corruption techniques, is that it will lead to a more robust and versatile pre-trained model which can perform competitively for a variety of downstream tasks, from sequence classification tasks to sequence generation tasks such as dialogue generation and abstractive question answering. After thorough examination using different combinations of alterations of the input for different tasks, the authors concluded that the success of the noising transformations depends on the downstream task. They found that the Text Filling objective, where random spans of consecutive tokens are masked instead of individual tokens, performed consistently well across all tasks and benefited greatly the performance on the summarization task. The final model is pre-trained using the Text Filling combined with the Sentence Permutation objective on the same data as RoBERTa [38].

2.3.2 Longformer Encoder Decoder (LED)

Pre-trained encoder decoder models used for sequence-to-sequence prediction that use full self attention, do not scale well for long input documents, as the resources needed for this operation grows quadratically with sequence length. Beltagy et al. [4] attempt to tackle this problem by modifying the attention mechanism so that it scales linearly with sequence length. They achieve this by implementing a *sparse* attention mechanism, through a *sliding window* operation of a specific size w, that attends to the local context of each word instead

of the whole sequence. In addition, an operation called *global attention* is used, so that certain tokens are attending to every other token in the sequence and vice versa. A visual representation of the differences between these attention mechanisms can be seen in Figure 3. This operation is useful to tailor the model for a specific task, for example for the summarization task, global attention can be applied to the BOS token, where in this way the whole sequence representations can be accumulated. The decoder of this model applies full attention to the encoder states as well as the previous decoder states. As far as pretraining is concerned, the authors copy the weights and architecture of the BART model, and in order to extend the input length, they expand the 1024 position embeddings to reach 16K, which are initialized by concatenating 16 times the 1K position embeddings of BART.



(a) Full n^2 attention (b) Sliding window attention (c) Dilated sliding window (d) Global+sliding window

Figure 3: From Beltagy et al. [4]: Comparing the full self-attention pattern and the configuration of attention patterns in the Longformer

2.3.3 PRIMERA

Xiao et al. [67] introduced a pre-trained language model for the multi-document summarization task with the goals of minimizing the need for dataset specific architectures and the need for a lot of fine-tuning labeled data. The latter goal is tackled through the pretraining process, which uses the Gap Sentence Generation objective (GSG) where the choice of which sentences to mask is based on their saliency. Specifically, the documents belong to clusters, so a salient sentence is a sentence that contains named entities that appear often in the cluster, and a sentence that has a high ROUGE overlap with other sentences in the same cluster. This method of masking sentences, is grounded on the intuition, that reconstructing these selected sentences will enable the model to learn to spot important information across documents of the cluster. The goal of limiting the data-specific modelling, is addressed by concatenating the cluster's documents into one sequence, separated by a special token, and making that to be the input of the model, alleviating the need for extensive preprocessing of the dataset (e.g. extraction of auxiliary information).

2.3.4 SUMMN2

Handling longer input documents is addressed with a different strategy in the $SUMM^N$ [73] summarizer. Instead of employing a sparse attention mechanism, they develop a framework consisting of multiple stages of abstractive summary generation where the input text is split based on the maximum sequence length the backbone model can handle as input. After the input text is split, the target summary is divided into sentences and a subset of them is matched to each of the segmented input pieces, forming a new source-target pair (first step of the coarse stage displayed in Figure 4). The matching is a heuristic process, where sentences of the target text are matched greedily with each source segment using the ROUGE-1 metric. Consequently, these generated pairs are fed to the coarse summary generation step of the framework (second step of the coarse stage displayed in Figure 4).

The purpose of the coarse stage, is to compress the input text of the dataset to a length where the backbone model in final fine-grained stage can handle. Therefore, there can be multiple coarse stages, where each of them treat as source text the generated abstractive summary of the previous coarse stage. The number of coarse stages can be estimated by the following equation:

$$\hat{N} = \lceil \frac{logK - logd_1}{logc_1 - logK} \rceil$$

,where K is the maximum sequence length of the backbone model, d_1 and c_1 is the average input source text length and the average coarse segment length in stage 1.

In principle, any abstractive transformer encoder decoder model can serve as a backbone model for the framework. The authors use BART for their implementation, as it is a model that performs well on shorter forms of text, thus the experiments can highlight how the $SUMM^N$ framework can extend the backbone model's capabilities to longer forms of input. Through experimentation, the authors conclude that by using different instances of the backbone model for the N stages, yields better results than using instances of the model that share the same weights.

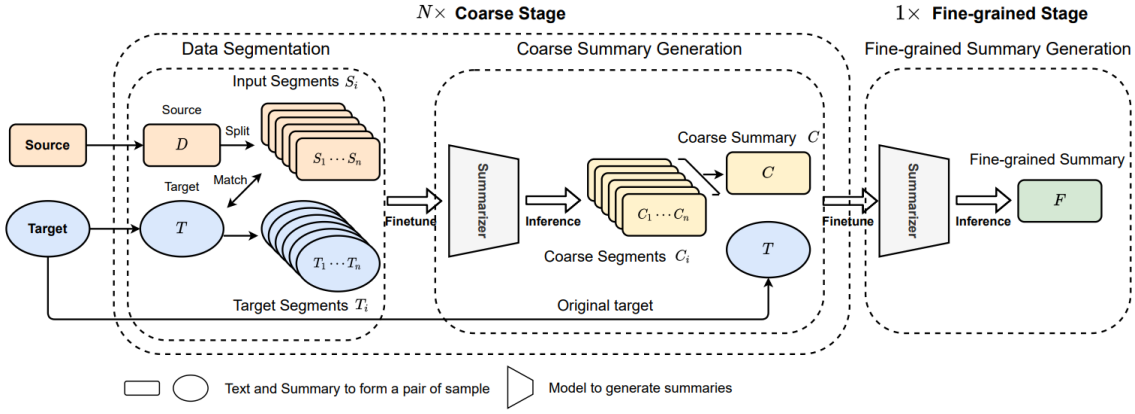


Figure 4: From Zhang et al. [73]: Workflow of the SUMMN framework. It contains N coarse stages and 1 fine-grained stage. At each coarse stage, source and target text is segmented and paired using a ROUGE-based greedy algorithm, and then a backbone summarization model is used to generate the summary for each segment. After multiple coarse stages, the last fine-graded stage produces the final summary output.

2.3.5 BART with Graph Encodings

BART with Graph Encodings [45] refers to an approach for multi-document summarization using auxiliary graph information as input, complementary to the source text input using BART as its backbone transformer encoder-decoder model. In the multi-document setting, a collection of different documents pertaining to one topic, namely a document cluster, are to summarized. In this work, the authors choose to simply concatenate these texts to form one single input. However, this leads to lengthy input, therefore each document gets truncated to L/N length where L is the maximum sequence length of the backbone model and N is the number of documents of a document cluster. Moreover, to alleviate from the hindrance of limited sequence length, the Longformer Encoder Decoder is employed – initialized with BART weights and architecture – which can accept as input more tokens due to its sparse attention mechanism. This model is referred to as BART-Long.

As far as the graph text input is concerned, the graph information is derived by performing co-reference resolution and open information extraction to each document of a document cluster. This results in subject-predicate-object triplets where subjects and objects define the nodes whereas the predicates define the edges of the graph. Furthermore, the TF-IDF metric is calculated for each word in a document, which can serve as an indicator of semantically close subject words or phrases, and additionally to the co-reference resolution results, words or phrases that refer to the same entity are grouped together under a unique string that represents a subject node of the graph. Afterwards, the graph is pruned by removing sub-graphs consisting only of two nodes. Finally, the graph is transformed to text, by sorting sub-graphs in descending order in regard to their size, and traversing them in a breadth-first manner starting from the most central node, concatenating their text. The special tokens <sub>, <pred>, <obj> and <cat> are injected in the text as well, to signify the role of the text contained between the special token and the next special token. An example of how the graph is constructed and how the final form of the linearized text looks like can be seen in Figure 5. The graph-text result is then encoded by a non-pretrained transformer encoder with Longformer layers. The output is concatenated with the output of the second-to-last layer of the pre-trained BART text encoder and are finally fed to the last layer of the text encoder. Subsequently, the text encoder output is used by the BART pre-trained decoder's attention mechanism to generate the summary.

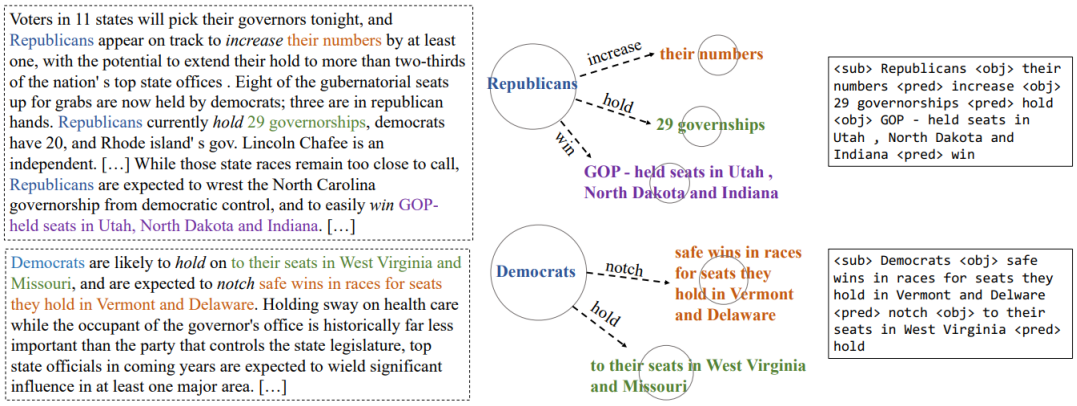


Figure 5: From Pasunuru et al. [45]: Example of how a graph is constructed for a couple of documents and of how the linearized graph text looks like.

Table 1: Type of attention used in the different models and their contribution to the summarization task.

Model	Attention of Model	Novelties
BART	Full n^2	Encoder-Decoder pretraining
LED	Sliding Window + Global	Handles longer input
PRIMERA	Sliding Window + Global	Pretraining objective tailored to Multi-Document Summarization
BART-graph	Sliding Window + Global	Encoding both text and linearized graph text information
SUMM-N	Full n^2	Multi-stage coarse-to-fine framework with abstractive intermediate stages

2.4 Evaluation of Generated Summaries

Summarizing a piece of text in an abstractive manner is an open-ended task, i.e. a task that has no singular correct answer, as there are multiple ways to express the same thing in natural language [36]. Therefore, evaluating the quality of a summary is not a trivial assignment even for a human [17].

Human assessment An automatically generated abstractive summary should be evaluated on the quality of its structure as well as its content. Assessment under the following four dimensions is usually performed [29]: *coherence*, *fluency*, *consistency* and *relevance*. Coherence refers to how comprehensive a summary is as a whole. The transition between sentences should be natural leading to an understandable and well-organized piece of text. Fluency is focused on the structure of each sentence, where correct grammar and proper word form usage is evaluated. Still a fluent and coherent summary should remain faithful on the source text and not include fabricated facts, a feature that corresponds to consistency. Finally, relevance considers the inclusion of important information to the final summary. These assessments need to be performed by humans, as they require deep understanding of semantics of natural language. However, human judgement requires a lot of resources and effort [35]. More than one evaluation should be performed on a summary, as humans can be inconsistent and biased [17]. Automated evaluation methods are needed as consistent comparison tools

between different implementations of summarization systems, as well as assisting tools during the development of the implementations.

Automatic evaluation metrics A type of automatic evaluation metrics for the summarization task is the *word overlap* metrics. Given a gold standard summary, i.e. human authored, these metrics score the similarity between the automatically generated one and the gold standard one, by counting the co-occurrences of *n-grams*. The term n-gram refers to a grouping of tokens i.e. one, two or n consecutive tokens. A lot of variations of these type of computation were introduced by Lin et al. [35], in a package called ROUGE, standing for Recall Oriented Understudy for Gisting Evaluations. *Recall*, *precision* and *f1* ROUGE scores can be calculated for a model generated text and a reference text (see equations 1).

$$\begin{aligned} ROUGE_{recall} &= \frac{\text{num of n-grams in ref and model}}{\text{num of n-grams in ref}} \\ ROUGE_{precision} &= \frac{\text{num of n-grams in ref and model}}{\text{num of n-grams in model}} \\ ROUGE_{f1} &= 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \end{aligned} \tag{1}$$

The most used ones in the literature are the ROUGE-1, referring to unigram overlap, ROUGE-2 for bigram overlap and ROUGE-L which measures the longest matching sequence of words.
Another type of metric, are the *semantic overlap* metrics, aiming to address the choice of different wordings for the same meaning. One prominent example, is the *Pyramid* method [44], a semi-automatic method, that attempts to identify pieces of text – no longer than a clause – called *Summarization Content Units* (SCUs) and assess the quality of a generated summary based on its inclusion of SCUs. The SCUs are identified by comparing multiple human summaries and extracting similar sentences, contained in each of them, that are subsequently divided to clauses.

Another approach that emerged along with deep learning architectures, is the *model based* approach, where a model is trained to evaluate the semantic similarity of pieces of text. One example of this approach is BERTScore [72] that leverages the power of pretraining. Tokens of a reference and a candidate text are mapped to contextual embeddings of BERT and afterwards the pairwise cosine similarity of these vectors is computed. The overall similarity score is a weighted sum of the highest cosine similarity score for each reference token, weighted by the *idf* score of the token, so that high frequency words such as 'the' or 'is' do not contribute a lot to the final score.

Factual Consistency The dimension where automatic evaluation of abstractive summarization is still lacking is factual consistency. The appearance of hallucinations in generated abstracts has been investigated by Maynez et al. [40]. They concluded that the majority of generated summaries contained non-faithful content to the source text. This phenomenon can have two manifestations: either content from the source is presented in the summary in such a way where the message of the source text is altered (*intrinsic hallucinations*), or information that do not exist in the source text is included in the generated abstract (*extrinsic hallucinations*). Human judgement is still crucial for this kind of evaluation as good performance in automatic measures does not strongly correlate with summary faithfulness.

2.5 Related Work

The conversation summarization task encapsulates a range of automatic summarization tasks that each has different characteristics such as the number of conversation participants, the means through which the conversation is held and the topic of the conversation. Namely, online discussions are a multi-party conversation, held online and can involve a wide variety of topics. Incorporating speaker [39] and discourse structure [20] information to guide the summarization has been explored in the literature mainly for dialogue summarization which usually encompasses the domain subtasks of meeting, chat and customer service summarization. In the domain of online discussions, the conversation does not have the same dynamic as these domains, as users can comment on their own time, most often with longer text and they usually do not interact with the majority of the participants of the conversation. Because of this, the way to leverage this kind of auxiliary information is not always straightforward [19].

Instead of discourse structure, some multi-document summarization systems use semantic information extracted from a document cluster in order to create cross-document structures (e.g. graphs) that can guide the summarization. Zhou et al. [74] model the relationships between entity clusters referenced among the documents to be summarized using a graph attention neural network, leveraging the learned attention weights in the decoding process. A similar strategy is employed by Cui et al. [12], but in this paper authors aim to capture the relation between topics and semantic units (i.e. words, documents). A topic-document graph encoder is built and the contextual representations of topics and semantic units are used in a two-step decoding process. However, these models use only pre-trained encoders therefore they require extensive training datasets due to their high capacity which is difficult since conversation summarization datasets with human curated summaries are expensive to create so they are rarer in the literature [28]. There have been some attempts for automatically creating conversation summarization datasets with the goal of pre-training or training from scratch neural summarization models. The RCS dataset [69], is a large (more than 11 million instances) pretraining dataset, comprised of Reddit discussions. The authors created synthetic summaries of the discussions, by appending the *title* of the post as well as the *lead comment* of the whole thread, meaning the comment that has the highest ratio of likes to dislikes. They took this approach using the intuition that these two pieces of text guide the direction of the conversation. Similar approach was employed by Tarnpradab et al. [58] and Tampe et al. [57] for creating extractive reference summaries. The downsides of utilizing this kind of datasets are the bias to one view expressed in the conversation and lower quality compared to a summary authored by a human.

Lastly, two other pre-trained conditional language generation models that have yielded good performance on the automated summarization task are `T5` [49] and `PEGASUS` [71]. Raffel et al. pre-trained `T5` on unsupervised objectives like span masking but also on supervised objectives like translation, classification, reading comprehension, and summarization. For each training example they concatenated to the input sequence the task that needed to be performed. On the other hand, `PEGASUS` the pre-training objective was built with specifically abstractive summarization in mind as the downstream task. We could have chosen either model as our backbone model for the methods that we adopted since both have versions which accept longer sequences [22] [46], but chose `BART` and `BART-long` as they were the preferred models from the methods' authors.

3 Methods

3.1 Datasets

NYT The dataset that we are using for finetuning the models is a subset of a conversational summarization dataset created by Fabbri et al. [19], specifically the news comments one. It consists of 500 instances of discussions under New York Times articles where each discussion is accompanied by some metadata concerning the article and a human curated summary. The comments and metadata derive from a publicly available dataset of NYT articles and their respective comments published between 2017 and 2018. The human gold standard summaries are sourced through an online crowd sourcing platform where the workers were instructed to curate abstractive summaries of at least 40 and at most 90 words long. They were encouraged to summarise common themes present in the discussion by using phrases such as 'Many commenters discuss..', 'A few commenters mention...' as well as including notable details if the word length limit permitted it.

SENSEI Additionally to the test subset of the NYT dataset, we also test the performance of the trained models on the SENSEI dataset [3]. SENSEI consists of 18 Guardian articles and the accompanied user comments. For every example, it includes 2 or 3 human curated summaries authored by different annotators, where for every one of them there is a unconstrained and a constrained version (where the constrained summaries have a 150-200 word limit). As with the NYT dataset, the annotators are encouraged to group and include the different issues and viewpoints that are present in the discussion. For this reason the resulting summaries resemble the ones that the trained summarizers generate, which made the SENSEI dataset a good candidate for automatic evaluation of the generated summaries. As it can be seen in table 2 the statistics of the SENSEI and the NYT dataset are different in terms of source and summary length and the ratio of comments per discussion , which intensifies the interest of testing the performance of the models on it. Moreover, the dataset includes information from different stages of the annotation process such as labeling comments with short texts that capture its content, grouping comments that involve similar themes or linking sentences of the final summary to the groups that they formed earlier. In this work we use the constrained summaries as gold standards and we do not utilize the "meta-annotations".

Table 2: Average statistics of the NYT and SENSEI datasets. Input length and summary length refers to the average number of tokens of the input and of the reference summary respectively. #Docs/example corresponds to the average number of comments per example.

	Input length	#Docs/example	Summary length
NYT	1624	16.95	79
SENSEI	5445	102.83	265.5

Preprocessing The dataset is offered by the authors in the form of six text files consisting of a *source* file and a *target* file for each of the *train*, *validation* and *test* splits. The train split amounts to 200 examples, the validation split to 50 examples and the test split to 250 examples. The source files contain the comments for each example accompanied with some relevant metadata. Specifically the *title* of the article the comments were posted under, some *keywords* pertaining to the article, a small *snippet* of the article and for each comment there is a *score* which indicates the number of upvotes it received. The target files contain the corresponding gold standard summaries.

For the purposes of our experiments, we disregard the metadata provided. For each example we concatenate the comments into a single string using a special token (`<doc-sep>`) to indicate the bounds of each comment. Then the comments and corresponding summaries are saved to a `JSON` file where each example constitutes a `JSON` object, resulting in three files – one for each split. We choose this format because it can be transformed seamlessly to a `huggingface Dataset` instance. We did not perform any cleaning to the data, as any `HTML` present was removed by the authors. None of the summarization implementations that we used required any additional preprocessing as all of the methods provided scripts for any specific processing they required. However it was required to write some small scripts or inject/alter few lines of code to train and test the methods with our dataset.

3.2 Experimental Setup

We tried to keep the training details as similar as possible to each other, following the authors' suggestions as well as guidelines for few-shot finetuning since the size of our dataset is limited. For `PRIMERA` we used the `huggingface` codebase and for all other implementations we used the `fairseq` codebase. We used the Adam optimizer with learning rate of 3×10^{-5} and we finetuned the number of training steps on validation results. Specifically, PRIMERA was finetuned for 500 total updates with 50 updates as warm up, BARTLong and SUMM-N for 200 updates with 20 updates as warm up and BARTLong-graph for 300 updates with 30 updates as warm up. All models were trained on a single RTX 3090 GPU of 24GB of RAM and gradient accumulation is applied so that we can simulate larger batch sizes that otherwise would not be possible. For decoding we used a beam size of 6, the minimum number of tokens for the generated summaries was set to 40 and the maximum to 90 for the NYT dataset and for the SENSEI dataset the minimum was set to 40 and the maximum to 240.

3.3 Evaluation

3.3.1 Automated evaluation

To evaluate the performance of the trained models we used two automated measures: `ROUGE` (specifically `ROUGE1`, `ROUGE2`, `ROUGEL`) and `BERTscore`. The implementation used for these metrics is the implementation included in the `evaluate` python library, developed by `huggingface` [1]. The ROUGE implementation is a python native implementation [2] that replicates the functionality of the original `perl` package [35]. For BERTScore we chose the pre-trained embeddings of the `microsoft/deberta-xlarge-mnli` model as it correlates best with the human evaluation according to the developers of the implementation [3]. We also included the weighting of the importance of tokens through their *idf* scores [72]. For both metrics we report the F1 scores.

[1] `https://huggingface.co/evaluate-metric`
[2] `https://github.com/google-research/google-research/tree/master/rouge`
[3] `https://github.com/Tiiiger/bert_score`

Moreover, we analyzed some statistics to capture the quality of the summaries in terms of coherence, fluency, consistency and relevance.

Redundancy While we were exploring the generated summaries, we encountered examples that included identical or nearly identical sentences within the same summary. The inclusion of redundant information hinders the coherence and comprehensibility of the text. To investigate this further and be able to compare the different models on how much they repeat themselves, we decided to compute the F1 BERTscore between all the pairs of sentences of a given summary. Then we aggregate these scores for a given summary by averaging and lastly we average all the individual summary scores for a given method resulting into a single score which we call *inter-sentence BERTscore*. The higher this final score is, the more redundant summaries a model generates. Additionally, we use two more measures for redundancy from Xiao and Carenini [68]. The first is the *type-token ratio*, i.e., *ratio of unique unigrams*, *bigrams* and *trigrams* present in a piece of text, where the lower the value the higher the redundancy. The second is the *normalized inverse of diversity* (NID), which is the normalized by length inverse of *diversity*, i.e., the entropy of unigrams in a text. The entropy refers to the Shannon entropy of the bag of words distribution of the given text. Higher values of NID indicate higher redundancy.

Consistency For consistency, we experimented with a natural language inference method called SUMMAC [30] that aims to detect inconsistencies in a generated summary leveraging the output of a entailment classification system. *Entailment classification* is the natural language task where given a unit of text called *hypothesis* (in our case the generated summaries), we need to classify whether it is entailed by, contradicting or is neutral to another unit of text called *premise* (in our case the source documents [7]). When there is no entailment between a hypothesis-premise pair means that there is no semantic evidence within the premise to support the hypothesis and therefore is factually inconsistent. The novelty of SUMMAC is that it computes the entailment probability by aggregating the entailment probabilities of the possible pairs of document sentences and summary sentences. This granularity allows to identify situations where a summary sentence is not entailed by any sentence of the document, therefore the entailment score of the whole summary should be lower. For our use case, we computed the probability that the summary is entailed by the discussion, by aggregating the entailment score of every pair of summary sentences and comments. We achieved this by using the option of the implementation to set the desired granularity to *paragraph* for the discussions and to *sentence* for the summaries. The authors introduce two different variants of the model, where both receive the summary to document entailment score matrix as input. The $\texttt{SUMMAC}_{\texttt{ZS}}$ variant keeps the maximum entailment score for each summary unit (depending on the granularity that can be a sentence, pair of sentences or a paragraph) to form a singular vector that is consequently distilled to a single value by averaging. The $\texttt{SUMMAC}_{\texttt{CONV}}$ leverages a convolutional layer to turn the matrix into a score vector, in order to mitigate the effect of extreme values present in the matrix. For our purposes we use the $\texttt{SUMMAC}_{\texttt{CONV}}$ variation as it is shown that it performs best. As far as other settings are concerned, we chose the default options of the framework which were deemed as the best performing by ablation studies of the authors. Specifically, the selected NLI model for entailment classification is a `BERT-Large` model finetuned on the MNLI [64] and VitaminC [52] datasets, and between the classification categories of *entailment*, *contradiction* and *neutral* of the NLI task only the entailment was considered to compute the pair matrix scores.

Abstractness In the literature it has been observed that abstractive summarization models often include in the generated summaries lengthy extracts from the source documents [47]. For this reason we want to compare how abstractive the models are. We calculated the percentage of common 4-gram sub-sequences of the generated summaries with their respective discussions. The lower these measures will be, we assume higher abstractness.

Summarizing dispersed source information Online discussion summarization, as we have already mentioned, can be perceived as summarizing a set of multiple documents that pertain to the same topic. A relevant summary would be one that would include salient information spread across the different source documents. To quantify the ability of the models to generate summaries with information from multiple documents of a cluster, we applied an automated measure developed by Wonhandler et al. [65]. Their motivation is to determine the smallest in size set of source documents that are required to cover the information included in the summary. For example, if just one document can cover all the information in the summary, it means that the summary poorly represents the cluster of documents. For this purpose, summaries and source documents are split on a higher granularity level than the one of sentences, the *proposition span* level (i.e. arguments

related with a predicate). Then every possible pair is fed to a neural binary classifier to determine if the propositions of the pair are aligned, meaning they convey similar information. After this step, subsets of the source documents with increasing cardinality $k, k \in [1, n]$ are greedily built, so that each subset covers the highest percentage possible of summary information. The proportion of summary propositions that are covered from propositions in documents in a subset, is the *coverage score* of the subset. Using the coverage scores for all the subsets, they compute the convergence speed to 1 (i.e. all the summary information is covered by the source documents in a subset). The slower the convergence, the higher is the area above the curve (AAC) of the function from number of documents to coverage ratio of the respective subset, meaning more documents are needed to cover the summary content. High scores are better in our case as they indicate that the summary contains information dispersed across multiple documents.

For our purposes, we limited the number of discussions used for the evaluation of each system, including the reference summaries, to 100 examples of the NYT dataset since the developed methodology includes a neural classifier and it was computationally expensive to evaluate on all 250 examples. We also used the default setting for the threshold of the neural classifier (0.5) and the maximum number of subsets (10).

Similarity of generated summaries All the models we used in our experiments employ BART as their backbone model. This motivates us to measure the similarity between the generated summaries. We use the same settings of the BERTscore implementation as we used in the model performance evaluation, and compare each pair of methods.

3.3.2 Qualitative Evaluation

We will be also taking a closer look in selected examples from the NYT dataset with the purpose of displaying a small sample to the reader and also assess how the quantitative results are reflected through qualitative inspection. We selected 4 discussions using a random seed $(= 2022)$. The example demonstration includes the title of the article and a small snippet of the article, where the snippet is meta-information extracted from the article's webpage and it consists of few sentences that describe the topic of the article. Then the generated summaries and reference summary are accompanied by a handful of comments that we think they reflect the direction of the conversation.

4 Results and Discussion

4.1 Automated Evaluation

4.1.1 Model Performance

The ROUGE scores and BERTscore of the models can be seen in Table 3 and 4 respectively. We observe that the `BARTLong` model performs better in terms of ROUGE scores in both the NYT and SENSEI dataset. The only exception to that is the ROUGE1 score for the SENSEI dataset where `SUMM-N` performs better and comes as very close second when it comes to ROUGE2 and ROUGEL. `BARTLong-graph` performs worse than `PRIMERA` for the NYT dataset but better for the SENSEI one. In terms of BERTscore, we see similar ranking for the NYT dataset as in the ROUGE measures and for the SENSEI dataset we see again `SUMM-N` performing best with `BARTLong` being close second and the other two models being further behind with `PRIMERA` performing slightly better.

`PRIMERA` and `BARTLong-graph` are models that use `BART-long` as their back-bone architecture and are modified to perform better in the multi–document setting. The fact that they consistently under–perform makes us suspect that the enhancements are not adequately suitable for the domain of online discussions. Specifically, even though the pretraining objective of `PRIMERA` theoretically seems to complement the task at hand, the pretraining corpus exclusively consists of data derived from the news article domain. This may have biased the model to a news article structure and in the multi–document setting, news article clusters tend to have different connections between them compared to clusters of comments.

For `BARTLong-graph` we can pinpoint two probable reasons for the decreased performance. Firstly, since the construction of the graph relies on steps that utilize automated neural co-reference resolution and automated neural open information triplet extraction, the generated output, that later acts as input to the summarization model, can introduce additional noise, especially in the online discussion domain where the source text does not always follow the correct grammar. Secondly, the graph encodings that are later merged with the text encodings, are not pretrained and are trained solely from the training subset of the NYT dataset.

The 250 examples may not be sufficient to train the graph encodings, which may ultimately add noise rather than useful signal to the model. As far as SUMM-N is concerned, the fact that it performs better for the SENSEI dataset may be attributed to the fact that it can utilize more efficiently the multi–stage process since the input documents are longer.

However, these automated measures only capture the similarity of the generated summaries to the reference summary and is important to analyze their performance under different perspectives as well to conclude if a model produces consistently better summaries than the others.

Table 3: ROUGE1/ROUGE2/ROUGEL f1 scores.

Method/Dataset	NYT	SENSEI
BARTLong	**33.15/8.17/19.44**	32.21/**5.63/16.31**
PRIMERA	31.18/7.02/19.14	23.16/4.67/13.12
SUMM-N	32.34/7.61/19.02	**33.64**/5.36/15.92
BARTLong-graph	31.22/6.99/18.36	25.90/4.50/14.47

Table 4: BERTScore f1 scores.

Method/Dataset	NYT	SENSEI
BARTLong	**57.11**	54.13
PRIMERA	55.82	50.71
SUMM-N	56.91	**54.93**
BARTLong-graph	54.02	50.19

4.1.2 Analyses of summarization qualities

How redundant are the generated summaries? In Tables 5 and 6 we see how the models perform in terms of redundancy when they are tasked to generate summaries for the NYT and SENSEI discussions respectively. What we observe is across all the metrics, SUMM-N is the least redundant model. For the SENSEI dataset the values of the unique grams and inverse diversity measures are relatively worse than the ones for NYT, which is due to the finding that longer texts tend to be more redundant [68]. Notable is also the observation that for the SENSEI dataset the models seem to be even less redundant than the reference summaries. This can be explained by the fact that the average generated summary length across all models is 121 tokens in comparison the 265 tokens of the reference summaries.

Table 5: Redundancy measures for generated summaries based on NYT discussions. Namely: average ratio of unique 1-grams, 2-grams and 3-grams, Normalized Inverse Diversity and intersentence BERTScore f1 value.

NYT	U1-gram%	U2-gram%	U3-gram%	NDI	Intersentence BERTScore f1
BARTLong	66.42	93.59	**99.97**	0.151	62.62
PRIMERA	66.03	86.77	91.35	0.158	64.84
SUMM-N	**70.14**	**96.01**	99.95	**0.135**	**60.77**
BARTLong-graph	62.63	91.27	99.96	0.170	64.94
Reference	72.24	95.60	98.86	0.127	62.93

How factually consistent are the summaries? To assess the factual consistency of the generated summaries, we computed the entailment probability score for each discussion-summary pair of the NYT dataset

Table 6: Redundancy measures for generated summaries based on SENSEI discussions. Namely: average ratio of unique 1-grams, 2-grams and 3-grams, Normalized Inverse Diversity and intersentence BERTScore f1 value.

SENSEI	U1-gram%	U2-gram%	U3-gram%	NDI	Intersentence BERTScore f1
BARTLong	57.60	89.69	99.89	0.177	62.72
PRIMERA	57.12	77.94	83.57	0.188	66.73
SUMM-N	**62.03**	**93.71**	**100.00**	**0.156**	**59.83**
BARTLong-graph	57.17	87.80	99.25	0.189	65.42
Reference	56.01	91.02	98.10	0.175	60.05

using the SUMMAC system. In Table 7 we see the average of these scores for every system as well as for the reference summaries. We see that three out of the four models surpass the scores of the reference summaries with PRIMERA ranking first. This observation does not align with the expectation that a human written summary should be the most factually consistent. After investigating the content of individual summaries that had high entailment score, we noticed that they were highly extractive with some of them even containing segments that are exact copies of the source documents. As a result, these high scores skew the averaged final score that we report on. However, these results may also indicate a weakness of NLI models to evaluate the consistency of abstractive summaries. The MNLI and VITC datasets on which the underlying NLI model of SUMMAC is trained, contain single-sentence premises, whereas in our case we handle multi-sentence premise [61]. Moreover, in abstractive summarization a hypothesis may only be entailed by the combination of multiple premises (for example the hypothesis *Some commentators agree on x but are conflicted on y*). Therefore, it is hard to arrive to sound conclusions about the consistency of the systems based on these findings.

Table 7: Average probability of the generated summaries being entailed by the source documents.

Model	$\text{SUMMAC}_{\text{CONV}}$
BARTLong	30.06
PRIMERA	33.53
SUMM-N	29.78
BARTLong-graph	26.78
Reference	28.39

How abstractive are the summaries? In Table 8 we see the average proportion of 4-grams that are extracted unaltered from the source documents into the generated summary for each of the examined models. The model that seems to exhibit most this behavior is SUMM-N with second being PRIMERA. BARTLong-graph seems to produce the most abstractive summaries, but considering its low ROUGE scores and BERTScore against the reference summary, accompanied by the fact that it measured to be one of the most redundant systems, this abstractness measure may correlate with lack of salient information within its generated summaries.

How competently do models include dispersed source information in their summaries? The dispersion scores in Table 9 show that BARTLong summaries tend to include information originating from more source documents. This score is even slightly better than the reference summary score which can be attributed to the reliance of the dispersion measure on a neural NLI model to compute the similarity between source and summary. As mentioned by the authors as well, the neural aligner might have more difficulty correctly determining the alignment of abstractive text. Since the reference summaries are more abstractive, its dispersion scores may include more noise and therefore be assigned a lower score. Additionally, we observe that BARTLong-graph has a considerably low score. As a multi-document summarization method, we would expect that BARTLong-graph would perform better. This low score may correlate with the fact that BARTLong-graph

Table 8: Proportion of 4-grams in the summaries that are extracted from the input.

Model	NYT	SENSEI
BARTLong	16.32	10.68
PRIMERA	19.40	17.97
SUMM-N	20.22	19.68
BARTLong-graph	4.15	1.93
Reference	1.76	2.07

has a high abstractiveness score. However, the reference summaries are also very abstractive and they do have such a low dispersion score. Other explanations of this low score, could be lack of relevant information or redundancy of information. Both can be corroborated by the low performance in terms of the BERTScore (Table 4) and redundancy scores (Table 5, 6) respectively.

Table 9: Dispersion (AAC) scores and standard deviation of the summaries generated by each method.

Model	AAC
BARTLong	**4.05 ± 3.51**
PRIMERA	2.73 ± 2.94
SUMM-N	3.68 ± 2.98
BARTLong-graph	0.79 ± 2.01
Reference	3.98 ± 4.63

Table 10: Similarity ranking between models based on the f1 BERTScore

Rank	Model Pair	BERTScore
1	BARTLong – SUMM-N	64.72
2	BARTLong – BARTLong-graph	63.47
3	BARTLong – PRIMERA	62.69
4	SUMM-N – PRIMERA	62.35
5	SUMM-N – BARTLong-graph	60.48
6	PRIMERA – BARTLong-graph	59.85

How similar are the summaries to each other? In Table 10 we see the similarity scores between the summaries that the models generate. Firstly compared to Table 4 we see that all the generated summaries have stronger similarity to each other rather than to the respective reference summaries. BARTLong and SUMM-N are the most similar to each other. In addition to the fact that they are the top performing models in terms of comparison to the reference summary, it isn't surprising that they contain semantically similar information. Nevertheless, the two models have a different attention mechanism so it is notable that they are the closest to resemblance. BARTLong-graph is dissimilar to SUMM-N and PRIMERA which may correlate to the fact that the latter ones are the most extractive models while BARTLong-graph is the least extractive one.

4.2 Qualitative Evaluation

The first example we're going to explore further can be seen in Table 11. The topic of the article revolves around aperitivo drinks (i.e., pre–meal drinks) and around the setting within they are enjoyed in Italy as well. The commentators in turn, describe their own experiences with aperitivo in Italy. Compared to the reference summary, the generated summaries include specific mentions of drinks and places that are present in the

discussion, rendering them more extractive. This is not necessarily an undesirable behavior for this kind of discussion as we could argue that the reader can gain more valuable information by having access to the actual recommendations of the user comments rather than a more abstract sentence like "some comments talk about the drinks that were served and that they said were wonderful". However this example shows the weakness of the automatic summarization systems to remain factually consistent. A type of inconsistency that is prevalent in this example is miscounting the number of commentators that have stated a specific view. We have highlighted with red the quantifiers that are inconsistent. Furthermore, there are inconsistencies with the named entities included in the summaries as well. The locations mentioned are not always precise and the type of a named entity occasionally is not the correct one (e.g. mistaking a drink as something that can be eaten). These instances are highlighted in the Table 11 with orange. As far as coherence and fluency are concerned, in this example we see that the models sometimes struggle with dangling anaphora, repeating information that were previously mentioned and ending the summary mid–sentence which can be confusing to the reader. We have highlighted these occurrences with teal.

The second example we are going to examine can be seen in Table 12. The article and the discussion center on the Winter Olympics taking place for the first time in Korea. In this example the generated summaries fall behind in comparison to the reference summary in terms of conveying in a coherent way the views in the discussion. We again observe problems with consistency and coherence. Additionally, the SUMM-N model has produced a highly extractive output, as the summary is the concatenation of two extracted sentences, which belong to a single comment, where the only alteration from the source material is the addition of "one/another commenter" in front of them. This not only limits the summary to a single view, but gives the false impression to the reader that these are comments authored by different people. The relevancy of the generated summaries suffer in this example, as we see that BART-long and PRIMERA include an irrelevant comment about Japan and the summary from BART-graph elaborates on a view about the difficulty of building ski resorts onto mountains, where in the original comment this is only a fraction of the text and is used to support the argument that building the winter sport facilities will not benefit Korea financially.

The third example can be seen in Table 13. The article's topic is the recollection of memories from a young age and the discussion consists of testimonies from users about how far back themselves can remember. In comparison with the reference summary, the generated summaries include individual commentators' experiences. The models' summaries have not captured the fact that some commenters disagree with the research cited from the article, that memory recollection is rare for events that happened before 3 or 4 years old. BART-long's summary includes that view but continues with a contradicting statement. Although both clauses are factually consistent with the discussion, they are not expressed by the same people. Also, the BART-graph summary contains extrinsic hallucinations, i.e., information that can not be referred back to the document. For example at no point in the discussion there is a mention of a "car show".

The fourth and final example can be seen in Table 14. The article discusses presidential biographies as a book genre and the author expresses what benefits they reap from reading them. The commenters express their own suggestions for presidential biographies and they compare events of the current political situation with past events described in the books. In this example the BART-graph summary is not relevant to the discussion at all. The are some connections through the mention of books, history and president, however in the comments there is no debate about the author and a president. As far as the other models are concerened, they all are successful in including the general sentiment of the discussion that reading presidential biographies can be useful for putting the modern political situation into perspective. Moreover, we observe a commonality as all summaries contain inconsistencies where a view has been attributed to the wrong entity. For example, Benjamin Franklin is mentioned multiple times throughout the conversation regarding the fact that he was not ever a president but the article featured his picture. In the summary produced by BART-long it states that a commenter read his biography which is not true. The same applies for the other two models, namely although the entities they include are part of the discussion, they are not used in the right context.

5 Discussion and Future Work

According to the quantitative and qualitative analysis no model consistently performs better than the others. BART-long and SUMM-N rank higher in the quantitative measures. We observe that the results of some quantitative measures are reflected through the qualitative analysis of the examples. The example of Table 12, is indicative of the last place of SUMM-N in terms of abstractiveness and the first three examples show the high redundancy of PRIMERA. All models contain factual inconsistencies. We observe two patterns: they often mistake the number of people that have expressed an opinion and they fuse information in a way that does not mirror

the semantic meaning of the source document. In this research we did not include any structural information about whether a comment was a reply to another comment (i.e., they form a thread). In the future, it would be interesting to investigate whether leveraging discourse information could aid alleviate the first pattern of errors. For example, attending to special tokens that signify discourse hierarchy (e.g. thread and comments) or special tokens for agreement and contradiction could help distinguish opinions that different parties of the conversation hold. Notably, `BARTLong-graph` includes hallucinations that cannot be traced back to the source documents. We speculate that `BARTLong-graph` may under-perform with the discussion dataset used because even though the model has a pre-trained encoder and decoder, the encoder of the cross-document graph is trained from scratch. Because of the small size of the training dataset this part of the model may add noise rather than guide effectively the summarization.

Based on the analysis, the long-input document methods produce better summaries. However, sometimes they are highly extractive, motivating future work where we can assess whether a different backbone model than `BART` can lead to less extractive output. Additionally, the cross-document techniques might need to be tailored to special characteristics of conversation in order to be able to add value to the summarization. Lastly, automatic summarization evaluation techniques for the multi-document setting or conversation domain are lacking in the literature so it would be an interesting future research avenue as well.

6 Conclusion

In this work we examined how Long single-document abstractive summarization and Multi-document abstractive summarization techniques perform on the domain of online discussions. Moreover, the limited availability of substantial datasets with human curated reference summaries for this domain, led us to focus on methods that use pre-trained language models. After automatic evaluation on different qualities and closer inspection of randomly selected examples, our conclusion to the question "How do recent pre-trained multi-document and long-input summarization techniques perform on the online conversation domain?" is that the Long Single-Document summarization methods performed better, however, both paradigms suffered from hallucinations and highly extractive output. Moreover, regarding the second research question "Can we pinpoint qualitative differences between them?", we observed higher redundancy and lower rate of incorporating information dispersed across documents. These observations are counter-intuitive as multi–document techniques specifically cater to that. Nevertheless, we believe that multi-document abstractive summarization is still under-researched in terms of dataset curation, approaches and evaluation methods in comparison to single-document summarization, so there is potential in researching further its application to the online discussion summarization task.

References

[1] SA Babar and Pallavi D Patil. Improving performance of text summarization. *Procedia Computer Science*, 46:354–363, 2015.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[3] Emma Barker, Monica Lestari Paramita, Ahmet Aker, Emina Kurtić, Mark Hepple, and Robert Gaizauskas. The sensei annotated corpus: Human summaries of reader comment conversations in on-line news. In *Proceedings of the 17th annual meeting of the special interest group on discourse and dialogue*, pages 42–52, 2016.

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

[5] Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. A neural probabilistic language model. *Advances in neural information processing systems*, 13, 2000.

[6] Sumit Bhatia, Prakhar Biyani, and Prasenjit Mitra. Summarizing online forum discussions–can dialog acts of individual messages help? In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 2127–2131, 2014.

[7] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[8] Sanyuan Chen, Yutai Hou, Yiming Cui, Wanxiang Che, Ting Liu, and Xiangzhan Yu. Recall and learn: Fine-tuning deep pretrained language models with less forgetting. 04 2020.

[9] Jackie Chi Kit Cheung and Gerald Penn. Unsupervised sentence enhancement for automatic summarization. In *EMNLP*, pages 775–786. Citeseer, 2014.

[10] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[11] Trevor Cohn and Mirella Lapata. Sentence compression beyond word deletion. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 137–144, 2008.

[12] Peng Cui and Le Hu. Topic-guided abstractive multi-document summarization. *arXiv preprint arXiv:2110.11207*, 2021.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[14] Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah A. Smith. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *ArXiv*, abs/2002.06305, 2020.

[15] Wafaa S El-Kassas, Cherif R Salama, Ahmed A Rafea, and Hoda K Mohamed. Automatic text summarization: A comprehensive survey. *Expert Systems with Applications*, 165:113679, 2021.

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

[17] Alexander R Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. Summeval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409, 2021.

[18] Alexander R Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir R Radev. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. *arXiv preprint arXiv:1906.01749*, 2019.

[19] Alexander R Fabbri, Faiaz Rahman, Imad Rizvi, Borui Wang, Haoran Li, Yashar Mehdad, and Dragomir Radev. Convosumm: Conversation summarization benchmark and improved abstractive summarization with argument mining. *arXiv preprint arXiv:2106.00829*, 2021.

[20] Xiachong Feng, Xiaocheng Feng, Bing Qin, and Xinwei Geng. Dialogue discourse-aware graph model and data augmentation for meeting summarization. In *International Joint Conference on Artificial Intelligence*, 2020.

[21] Shima Gerani, Yashar Mehdad, Giuseppe Carenini, Raymond Ng, and Bita Nejat. Abstractive summarization of product reviews using discourse structure. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1602–1613, 2014.

[22] Mandy Guo, Joshua Ainslie, David Uthus, Santiago Ontanon, Jianmo Ni, Yun-Hsuan Sung, and Yinfei Yang. LongT5: Efficient text-to-text transformer for long sequences. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 724–736, Seattle, United States, July 2022. Association for Computational Linguistics.

[23] Vishal Gupta and Gurpreet Singh Lehal. A survey of text summarization extractive techniques. *Journal of emerging technologies in web intelligence*, 2(3):258–268, 2010.

[24] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28, 2015.

[25] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[26] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[27] Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. Efficient attentions for long document summarization. *arXiv preprint arXiv:2104.02112*, 2021.

[28] Misha Khalman, Yao Zhao, and Mohammad Saleh. Forumsum: A multi-speaker conversation summarization dataset. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4592–4599, 2021.

[29] Wojciech Kryściński, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. Neural text summarization: A critical evaluation. *arXiv preprint arXiv:1908.08960*, 2019.

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

[31] Logan Lebanoff, Kaiqiang Song, Franck Dernoncourt, Doo Soon Kim, Seokhwan Kim, Walter Chang, and Fei Liu. Scoring sentence singletons and pairs for abstractive summarization. *arXiv preprint arXiv:1906.00077*, 2019.

[32] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

[33] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. *Advances in neural information processing systems*, 31, 2018.

[34] Wei Li, Xinyan Xiao, Jiachen Liu, Hua Wu, Haifeng Wang, and Junping Du. Leveraging graph to improve abstractive multi-document summarization. *arXiv preprint arXiv:2005.10043*, 2020.

[35] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

[36] Chin-Yew Lin and Eduard Hovy. Manual and automatic evaluation of summaries. In *Proceedings of the ACL-02 Workshop on Automatic Summarization*, pages 45–51, Phildadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.

[37] Peter J Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. Generating wikipedia by summarizing long sequences. *arXiv preprint arXiv:1801.10198*, 2018.

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

[39] Zhengyuan Liu and Nancy F. Chen. Controllable neural dialogue summarization with personal named entity planning. In *Conference on Empirical Methods in Natural Language Processing*, 2021.

[40] Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. On faithfulness and factuality in abstractive summarization. *arXiv preprint arXiv:2005.00661*, 2020.

[41] Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Thirty-first AAAI conference on artificial intelligence*, 2017.

[42] Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*, 2016.

[43] Shashi Narayan, Shay B Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *arXiv preprint arXiv:1808.08745*, 2018.

[44] Ani Nenkova, Rebecca Passonneau, and Kathleen McKeown. The pyramid method: Incorporating human content selection variation in summarization evaluation. *ACM Transactions on Speech and Language Processing (TSLP)*, 4(2):4–es, 2007.

[45] Ramakanth Pasunuru, Mengwen Liu, Mohit Bansal, Sujith Ravi, and Markus Dreyer. Efficiently summarizing text and graph encodings of multi-document clusters. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4768–4779, 2021.

[46] Jason Phang, Yao Zhao, and Peter J. Liu. Investigating efficiently extending transformers for long input summarization. *ArXiv*, abs/2208.04347, 2022.

[47] Jonathan Pilault, Raymond Li, Sandeep Subramanian, and Christopher Pal. On extractive and abstractive neural document summarization with transformer language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9308–9319, 2020.

[48] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[49] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

[50] Zhaochun Ren, Jun Ma, Shuaiqiang Wang, and Yang Liu. Summarizing web forum threads based on a latent topic propagation process. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 879–884, 2011.

[51] Alexander M Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*, 2015.

[52] Tal Schuster, Adam Fisch, and Regina Barzilay. Get your vitamin C! robust fact verification with contrastive evidence. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 624–643, Online, June 2021. Association for Computational Linguistics.

[53] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015.

[54] Eva Sharma, Luyang Huang, Zhe Hu, and Lu Wang. An entity-driven framework for abstractive summarization. *arXiv preprint arXiv:1909.02059*, 2019.

[55] Sajad Sotudeh, Hanieh Deilamsalehy, Franck Dernoncourt, and Nazli Goharian. Tldr9+: A large scale resource for extreme summarization of social media posts. *arXiv preprint arXiv:2110.01159*, 2021.

[56] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014.

[57] Ignacio Tampe, Marcelo Mendoza, and Evangelos Milios. Neural abstractive unsupervised summarization of online news discussions. In *Proceedings of SAI Intelligent Systems Conference*, pages 822–841. Springer, 2021.

[58] Sansiri Tarnpradab, Fereshteh Jafariakinabad, and Kien A Hua. Improving online forums summarization via hierarchical unified deep neural network. *arXiv preprint arXiv:2103.13587*, 2021.

[59] Sansiri Tarnpradab, Fei Liu, and Kien A Hua. Toward extractive summarization of online forum discussions via hierarchical attention networks. In *The Thirtieth International Flairs Conference*, 2017.

[60] Ian Tenney, Dipanjan Das, and Ellie Pavlick. Bert rediscovers the classical nlp pipeline. *arXiv preprint arXiv:1905.05950*, 2019.

[61] Prasetya Utama, Joshua Bambrick, Nafise Moosavi, and Iryna Gurevych. Falsesum: Generating document-level NLI examples for recognizing factual inconsistency in summarization. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2763–2776, Seattle, United States, July 2022. Association for Computational Linguistics.

[62] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[63] Suzan Verberne, Emiel Krahmer, Iris Hendrickx, Sander Wubben, and Antal van Den Bosch. Creating a reference data set for the summarization of discussion forum threads. *Language Resources and Evaluation*, 52(2):461–483, 2018.

[64] Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. 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 1112–1122, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[65] Ruben Wolhandler, Arie Cattan, Ori Ernst, and Ido Dagan. How "multi" is multi-document summarization? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5761–5769, December 2022.

[66] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

[67] Wen Xiao, Iz Beltagy, Giuseppe Carenini, and Arman Cohan. Primer: Pyramid-based masked sentence pre-training for multi-document summarization. *arXiv preprint arXiv:2110.08499*, 2021.

[68] Wen Xiao and Giuseppe Carenini. Systematically exploring redundancy reduction in summarizing long documents. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 516–528, Suzhou, China, December 2020. Association for Computational Linguistics.

[69] Ze Yang, Liran Wang, Zhoujin Tian, Wei Wu, and Zhoujun Li. Tanet: Thread-aware pretraining for abstractive conversational summarization. *arXiv preprint arXiv:2204.04504*, 2022.

[70] DING YING and Jing Jiang. Towards opinion summarization from online forums. ACL, 2015.

[71] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR, 2020.

[72] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.

[73] Yusen Zhang, Ansong Ni, Ziming Mao, Chen Henry Wu, Chenguang Zhu, Budhaditya Deb, Ahmed H Awadallah, Dragomir Radev, and Rui Zhang. Summˆ n: A multi-stage summarization framework for long input dialogues and documents. *arXiv preprint arXiv:2110.10150*, 2021.

[74] Hao Zhou, Weidong Ren, Gongshen Liu, Bo Su, and Wei Lu. Entity-aware abstractive multi-document summarization. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 351–362, 2021.

[75] Chenguang Zhu, Ruochen Xu, Michael Zeng, and Xuedong Huang. A hierarchical network for abstractive meeting summarization with cross-domain pretraining. *arXiv preprint arXiv:2004.02016*, 2020.

Table 11: Sample of discussion about commentator experiences with aperitivo and the respective generated summaries. Factual inconsistencies have been highlighted in red and orange. The red ones indicate a wrong use of quantifier for the number of commenters that expressed a view and the orange ones indicate general inconsistencies. Coherence and fluency issues are highlighted in teal.

Headline : *Take It Slow*
Snippet : *Partaking in aperitivo time, that easy stretch of sipping and snacking.*
Statistics : `Number of words in discussion:` 587, `Number of comments:` 11
Comment : Nothing like the secreted strong drink, to get you in the mood, or out of the mood - for what? before you get on the xxx train from Grand Central!
Comment : Had a rhumbero and Coca Cola at Cara Mia in Soth Beach last night. Seems to fit the bill! :-)
Comment : We stayed at the elegant Inn at the Spanish Steps in early November, where we enjoyed magnificent Negronis accompanied by a delicious selection of snacks, served on their lovely roof terrace.
Comment : This past week in between sessions of a film festival, a group of us did exactly this. And what a pleasure it was.Plonk in downtown Missoula, Montana isn't quite Rome but our Aperitivo was delighful nonetheless. Can't wait for warmer weather so we can move it outside.
Comment : When you mentioned "an elegant hotel bar in Rome" at the beginning of your article, exactly the Stravinskij at the Hotel de Russie came to my mind!Absolutely beautiful and well worth a visit, particularly in warmer weather when one can sit outside. It is near Piazza del Popolo...
Comment : Italian happy hours are wonderful. Typically, the price of a drink goes UP, not down, as in the US. However this covers the cost of the inevitable delightful food spread. It's wonderful to sit outside and sip, nosh and watch the world go by.
...

`Reference`	`BARTLong`	`PRIMERA`
Commentators talk about the tourist and gastronomic experiences in Italy. Some commentators cited the beauty of the hotel they stayed at. Other commentators cite his excellent experience during an Italian Happy Hour. Some comments talk about the drinks that were served and that they said were wonderful.	Several commenters talk about their experiences with happy hours in Italy. One commenter talks about an elegant hotel bar in Rome called the Stravinskij, while another talks about a restaurant in the Mediterraneo Hotel in Montana called the Hotel Locarno. A couple of commenters discuss their experiences at aperitivo's in Missoula, Montana. A few other commenters share stories about their happy hours experiences in Italy and how they enjoyed them	Many commenters share their experiences with the Spritz in Rome, ranging from an elegant hotel bar to a rhumbero and Coca-Cola at Cara Mia in Soth Beach. A couple of commenters recommend visiting the Hotel Locarno off Piazza di Popoli. One commenter talks about his experience with the Spritz in Rome, and another says they enjoyed it in Montana.

`SummN`	`BARTGraph`
Most of the commenters share their experiences of enjoying aperitivo in Rome and elsewhere in Italy, with some giving details of their favorite bars and restaurants. A few commenters mention their experiences with Negroni in the Mediterraneo Hotel in Rome, while another talks about Cynar in the Stravinskij bar near Piazza del Popolo. A couple of commenters talk about their experience with rhumbero and Coca Cola at	Several commenters share their experiences of eating and drinking Negroni in Rome. One commenter says that the restaurant is beautiful and well worth a visit, while another says that there is no room to sit outside in the warm weather and that they can't wait for warmer weather so they can move it outside. Another commenter talks about their experience at the Hotel Stinskij, a restaurant and bar in the Spanish Steps.

Table 12: Sample of discussion about Winter Olympics taking place for the first time in Korea and the respective generated summaries. Factual inconsistencies have been highlighted in red and orange. The red ones indicate a wrong use of quantifier for the number of commenters that expressed a view and the orange ones indicate general inconsistencies. Coherence and fluency issues are highlighted in teal. The SUMM-N summary is not highlighted at all because it is a concatenation of two extracted sentences from the source document.

Headline : *New Slopes*
Snippet : *South Korea is a skiing afterthought no more. The 2018 Winter Olympics is a coming-out party for the country as a winter sports destination.*
Statistics : `Number of words in discussion:` 731, `Number of comments:` 13
Comment : They mowed down old growth forests with protected species to build the ski runs for the Olympics. Such things wouldn't have been allowed in the US or the west. International events like Olympics for Koreans are still nationally organized efforts to show off their national pride and they are paying huge, irreparable prices for it.
Comment : The skiing in Hokkaido, Japan looks amazing. So much powder. I want to go there sometime.
Comment : Judging by the snow (or actually the lack of it) in these photos, I can only assume that having the Olympics there has nothing to do with winter sports. Rather, it's just another political exploitation, not unlike Sochi.
Comment : Most Koreans I spoke with are vehemently against the Olympics being hosted in their country. They view it as another sour mark in the long line of corruption by the right-wing party in nearly a decade's rule. Alpensia has been losing $55 million annually since it opened in 2011, and it's unbelievable the amount of money thrown to make it meet Olympic standards (and into the pockets of government officials and chaebols)...
Comment : When I was a young US soldier in South Korea in the 1960s, Korea had yet to discover indoor plumbing ... And now they have multi-million dollar ski resorts! How times have changed.
...

`Reference`	`BARTLong`	`PRIMERA`
Several commenters agree that the Olympics, especially the Winter Olympics, can lead to exploitation of a country's resources by politicians, so they are not particularly excited by the Olympics. They believe it was very costly to be able to build ski resort venues in areas where snow may not naturally occur and that there will be few benefits besides national pride. A few commenters were expressing their enthusiasm at how far Korea has developed since the Korean War.	The commenters discuss how the skiing in Hokkaido, Japan, looks amazing and how the Olympics are being hosted in their country despite the fact that there is no snow there. Several commenters also discuss the political exploitation of the Olympics in Korea by the right-wing government and how it backfires in the long run. One commenter points out that the Winter Olympics, especially skiing, is never global as there are no winter in many parts of the world	Many commenters believe that the Olympics in Korea have nothing to do with winter sports and that it is just another political exploitation, similar to that in Sochi. A couple of commenters mention that the skiing in Hokkaido, Japan, is amazing. One commenter implies that the Olympics in Korea have nothing to do with winter sports and that it is just another political exploitation, similar to that in Sochi .

`SummN`	`BARTGraph`
One commenter points out that Alpensia has been losing $55 million annually since it opened in 2011, and it's unbelievable the amount of money thrown to make it meet Olympic standards (and into the pockets of government officials and chaebols). Another commenter talks about how Koreans are vehemently against the Olympics being hosted in their country, and they view it as another sour mark in the long line of corruption by the right-wing party.	Most commenters seem to agree that skiing in Korea is not as good as it used to be, with one commenter saying that it's gotten worse, while another says that it has gotten better . A couple of commenters talk about the difficulty of building ski resorts in Korea, especially in the mountains, and the amount of money that is thrown into them. One commenter talks about the difficulties of getting permits to build ski resorts, and another talks about how it

Table 13: Sample of discussion about recollection of memories. Extrinsic hallucinations of BART-graph have been highlighted in orange. Coherence and fluency issues are highlighted in blue.

Headline : *What Is Your Earliest Memory?*
Snippet : *How far back can you recall events in your own life and how old were you when you first remembered them?*
Statistics : Number of words in discussion: 3066, Number of comments: 25
Comment : .. I agree with the research that suggests ... Although I am able to remember bits and pieces of my earliest past events without looking through photographs, I recall the most when I discuss past events with my parents. ... When I was younger, I lived near Toys R Us. I always wanted to be there to the point where I would refuse to eat dinner at home. ... I find it surprising how I am able to recall events that happened when I was four, but not younger without discussing with my parents.
Comment : Yes, I agree with the research that this article is showing us. I think that there is a connection between brain memory and speaking, whether it be in sign language or English. ... 3. My oldest memory was when I was two years old. I can still remember it now, but I assume I thought about it when I was younger as well.
Comment : I do agree with the information stated, because I talk about memories of my childhood with my parents, and the information stated in through and accurate, ... My earliest memory was me drinking a 16 ounce bottle of formula in my little rocking chair, I was watching the Wiggles on my T.V, And I remember being sick that day as well.
Comment : I was less than three. I remember everyone running around and shouting and holding little us flags. everyone seemed happy. I learned later it was v j day. I did see a photo of us sitting on the porch at our sumner house. But I remember the noise and all people running around and shouting.
...

Reference	BARTLong	PRIMERA
One commenter says he agrees that we can't remember when we were very small, but he believes that if our parents tell us about an event that occurred during that time, we can remember something. Several commenters agree with the article, say they remember things that happened only after they were four years old and report the first memories they have. Others say they remember some events that occurred when they were less than four years old.	Most of the commenters share their earliest memories from when they were young, ranging from infancy to the age of 3 or 4. One commenter states that their earliest memory was when they went to Hawaii with their parents. Another commenter says that they can't remember anything before that. A couple of commenters disagree with the research and say that the earliest memories they can recall are from when their parents discuss past events with them. A few commenters talk about their earliest	Most of the commenters talk about their earliest memories when they were young. They remember things that they have never discussed with their parents. Some commenters recall events that happened when they were 4 years old, while others say that they remember it only when they discuss it with their parents. A few commenters talk about their earliest memories when they were young. They remember things that they have never discussed with their parents.

SummN	BARTGraph
Most of the commenters recall their earliest memories of being a child and how they can't remember anything before the age of 3. They recall the earliest memory they can remember was when they were four years old, and they recall it because they wanted to go to Toys R Us to play with the toys and because they would refuse to eat dinner at home. One commenter recalls being fed in a high-chair and having a 16-ounce bottle of formula	Several commenters reminisce about their earliest memories of childhood. One commenter recalls a time when they were young and their parents took them to a car show. Another commenter talks about the importance of remembering your earliest memories, especially those from when you were young. A few commenters talk about their own earliest memories and how they relate them to the article. One commenter reminisces about when he was young and his parents took him to a movie

Table 14: Sample of discussion about presidential biographies. Factual inconsistencies have been highlighted in red (wrong quantifiers) and orange (general inconsistencies). Coherence and fluency issues are highlighted in teal. The `BARTGraph` summary is fully highlighted as it has no connection to the discussion other than the mention of books and history.

Headline : *Presidential Biographies*
Snippet : *These books don't tell you that everything is going to be O.K. but rather that nothing was ever really O.K. to begin with.*
Statistics : `Number of words in discussion:` 2652, `Number of comments:` 28
Comment : Being depressed by our current president, I began reading ... I would also strongly suggest Ron Chernow's Grant as he is IMHO our least understood president and was one of our best.
Comment : Presidential biographies are my favorite way to read about U.S. history. If more Americans read them, I doubt Trump would be president.
Comment : Robert A. Caro's trilogy on Lyndon Johnson (Master of the Senate, The Path to Power, and Means of Ascent) are unparalleled. Beyond presidents, I would have to add Eleanor Roosevelt's autobiography...
Comment : Currently, I'm reading Avlon's book on Washington's Farewell Address. While Washington's writings will never compare to those of Jefferson, Franklin, Madison, Hamilton or Paine, the advice and observations in his address are very resonant today...
Comment : One of the presidential biographies pictured is about Benjamin Franklin, who, while certainly a founding father, was never president.
Comment : Trump's election left me feeling adrift, unhinged. Like 911. I was annoying my friends, ranting and frightened. I desperately needed perspective ... I recommend reading presidential biographies to all as a way of staying sane in the presence of a seemingly insane presidency.
...

`Reference`	`BARTLong`	`PRIMERA`
Several of the commenters also express the benefits of reading presidential biographies because of the helpful and informative perspective they give to current political events. One commenter suggests broadening the scope to include biographies of others, including presidents' wives. Another suggests that reading presidential biographies could actually be depressing because of their tarnished characters.	Most of the commenters talk about the importance of reading biographies of presidents, especially those who have been in office for a long time. Some talk about how much they enjoyed reading them and how they helped them stay sane in the midst of all of the craziness of the current administration. One commenter talks about reading Benjamin Franklin's biography and how it helped him stay sane during his presidency. A couple of commenters discuss how they would like to read more	The commenters discuss their experiences reading presidential biographies and how it has helped them to stay sane in the presence of a seemingly insane presidency. Some commenters say that while Washington's writings will never compare to those of Jefferson, Madison, Hamilton, or Paine, the advice from his Farewell Address is still relevant today. Other commenters talk about Eleanor Roosevelt's autobiography and how it has helped them to stay sane.

`SummN`	`BARTGraph`
Many of the commenters discuss the importance of reading biographies of U.S. presidents in general and specifically that of the current president, Donald Trump. Several commenters praise Ron Chernow for his biography of Grant, saying that he is one of the best biographers in the world. A couple of commenters suggest reading about Andrew Jackson and Grant. A few commenters talk about their own experiences reading about presidents and how they relate to Trump and his administration.	Commentators are split between supporting the author and the president. Those who support the president say that he is a good man, while those who dislike him think that the book is a "joke." Those who dislike the book say that the author is ignorant of history and has not done enough research. One commentator says that he did not enjoy the book, but that he will continue to read it because it is a great read.