Evaluating the Tradeoff Between Abstractiveness and Factuality in Abstractive Summarization

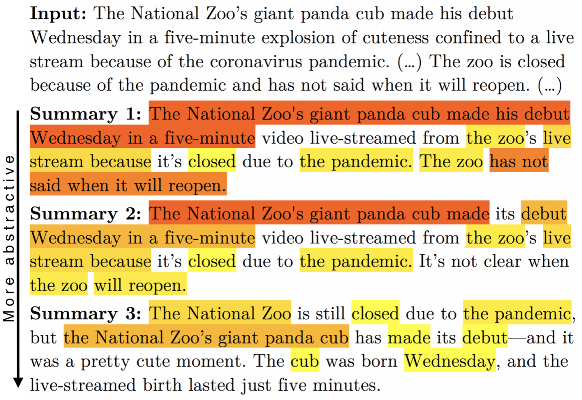
Markus Dreyer[1] **Mengwen Liu**[1] **Feng Nan**[1] **Sandeep Atluri**[1] **Sujith Ravi**[2]*
Amazon[1] SliceX[2] AI
{mddreyer, mengwliu, nanfen, satluri}@amazon.com
ravi.sujith@gmail.com

Abstract

Neural models for abstractive summarization tend to generate output that is fluent and well-formed but lacks semantic faithfulness, or factuality, with respect to the input documents. In this paper, we analyze the tradeoff between abstractiveness and factuality of generated summaries across multiple datasets and models, using extensive human evaluations of factuality. In our analysis, we visualize the rates of change in factuality as we gradually increase abstractiveness using a decoding constraint, and we observe that, while increased abstractiveness generally leads to a drop in factuality, the rate of factuality decay depends on factors such as the data that the system was trained on. We introduce two datasets with human factuality judgements; one containing 10.2k generated summaries with systematically varied degrees of abstractiveness; the other containing 4.2k summaries from five different summarization models. We propose new factuality metrics that adjust for the degree of abstractiveness, and we use them to compare the abstractiveness-adjusted factuality of previous summarization works, providing baselines for future work.[1]

1 Introduction

Summarization is the task of generating a semantically faithful, well-formed and concise text representation of the input. Automatically generated summaries have traditionally been *extractive* (Luhn, 1958; Edmundson, 1969; Neto et al., 2002; Erkan and Radev, 2004; Wong et al., 2008), leading to issues with readability and coherence, as different extracted fragments may not fit well when taken out of their original contexts (Poibeau and Saggion, 2012). Researchers have also invested in methods for *abstractive* summarization, aiming to paraphrase the input documents' main points without borrowing their exact lexical expressions (Radev and McKeown, 1998; Saggion and Lapalme, 2002; Ganesan et al., 2010; Genest and Lapalme, 2012; Radford et al., 2019; Gehrmann et al., 2019; Lewis et al., 2019; Zhang et al., 2020). Abstractive summaries generated by today's neural models tend to be fluent and well-formed, but lack semantic faithfulness (Cao et al., 2017; Kryscinski et al., 2019). Observed rates of factual errors in abstractive summaries have ranged from 30% to over 75% (Cao et al., 2017; Maynez et al., 2020). The research community is developing automatic factuality metrics (Wang et al., 2020; Kryscinski et al., 2020; Goodrich et al., 2019; Goyal and Durrett, 2020; Ribeiro et al., 2022) and methods that attempt to increase factuality (Fan et al., 2018; Scialom et al., 2019; Zhang et al., 2019; Falke et al., 2020; Cao and Wang, 2021). However, the factuality problem of abstractive summaries cannot be well understood without considering the *degree* of abstractiveness of a given summary: Any summary is on a spectrum between *extractive* and


Input: The National Zoo's giant panda cub made his debut Wednesday in a five-minute explosion of cuteness confined to a live stream because of the coronavirus pandemic. (...) The zoo is closed because of the pandemic and has not said when it will reopen. (...)

Summary 1: The National Zoo's giant panda cub made his debut Wednesday in a five-minute video live-streamed from the zoo's live stream because it's closed due to the pandemic. The zoo has not said when it will reopen.

Summary 2: The National Zoo's giant panda cub made its debut Wednesday in a five-minute video live-streamed from the zoo's live stream because it's closed due to the pandemic. It's not clear when the zoo will reopen.

Summary 3: The National Zoo is still closed due to the pandemic, but the National Zoo's giant panda cub has made its debut—and it was a pretty cute moment. The cub was born Wednesday, and the live-streamed birth lasted just five minutes.

Figure 1: Three successively more abstractive summaries generated from the same input article, with MINT abstractiveness scores (Section 2.1) of 46.1%, 67.2%, 79.5%. Fragments extracted from the input are marked from red (longer fragments) to yellow (shorter fragments). The bottom summary has factual errors.

*Work conducted during his position at Amazon.

[1] Code and data are available at https://github.com/amazon-science/abstractive-factual-tradeoff.

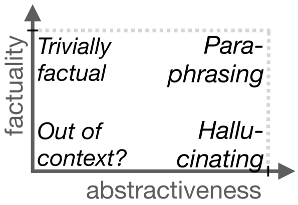


Figure 2: Four extremes at the abstractiveness-factuality spectrum.

abstractive (See et al., 2017). Summaries that are extractive to a larger extent tend to be more factual since copying text from the input into the summary rarely introduces factual errors while the task of paraphrasing, which results in summaries that are more *abstractive*, is harder and prone to semantic errors. As an example, Figure 1 shows part of a Washington Post article and three summaries with increasing abstractiveness, which we have generated using our abstractiveness constraints (Section 2.2). The first two summaries are correct, but the third, most abstractive, summary has factual errors, misinterpreting the input.

Few authors have discussed this connection explicitly. Lebanoff et al. (2019) observe that abstractive summaries consisting of concatenated extracted fragments tend to be more factual than those created by more complex fusion. Durmus et al. (2020) observe that models trained on the more *extractive* CNN/DM dataset (Hermann et al., 2015) create more factual summaries than models trained on the more *abstractive* XSum dataset (Narayan et al., 2018). We show that such models differ in factuality even when we bias them to generate summaries that have similar levels of abstractiveness. Our analysis (Section 4) situates summarization models on the spectrum outlined in Figure 2, where factual summaries range from “trivially factual” (extractive) to truly “paraphrasing” (abstractive). We make the following contributions:

1. We systematically explore the relationship of abstractiveness and factuality and show how factuality decays with increasing abstractiveness. We argue that factuality rates of different systems cannot be compared without taking their degrees of abstractiveness into account.
2. We introduce new factuality metrics that take abstractiveness into account and evaluate the abstractiveness-factuality tradeoff across various datasets and summarization models. We establish baselines that will allow others to demonstrate progress on mitigating the abstractiveness-factuality tradeoff.
3. We introduce a new dataset containing 10.2k summaries with systematically varied degrees of abstractiveness along with human factuality judgements, and a second dataset containing 4.2k summaries from five summarization models with their human factuality judgements.

2 Abstractiveness

2.1 Measuring Abstractiveness

In this paper, we wish to analyze the relationship of abstractiveness and factuality of generated summaries. We start by proposing a comprehensive abstractiveness metric. Abstractiveness measures the amount of rephrasing, i.e., the degree to which the words, phrases and sequences of the generated text have *not* been extracted from the corresponding input; a fully abstractive summary method expresses the main points of the input in its own words. To measure abstractiveness, most authors list the proportions of summary n-grams of varying lengths that are novel, i.e., do not occur in the corresponding inputs (See et al., 2017; Narayan et al., 2018; Gao et al., 2019). Grusky et al. (2018) proposed a new metric also based on contiguous overlapping text spans, *density*, measuring the average length of extracted fragments in a summary. Others have proposed metrics that take common *non-contiguous* subsequences into account, e.g., *perfect fusion*$_k$ (Durmus et al., 2020) measures the percentage of summary sentences that assemble substrings from k source sentences in their original order.

Based on these previous works, we define a comprehensive abstractiveness metric that combines measures of contiguous and non-contiguous extractive summary fragments, making it sensitive to different kinds of abstractiveness and therefore suitable as a general abstractiveness metric. We define this metric as a ratio, in order to facilitate combining it with a factuality metric of the same [0,1] range (Section 4). Let $\chi(\boldsymbol{x}, \boldsymbol{y}) = \mathrm{hmean}(p_1, p_2, p_3, p_4, \mathrm{lcsr})$ be a measure of *extractive* overlap between input $\boldsymbol{x}$ and summary $\boldsymbol{y}$, using the harmonic mean of multiple component measures. Each p_n, short for $p_n(\boldsymbol{x}, \boldsymbol{y})$, is the n-gram precision of the n-grams in $\boldsymbol{y}$ with respect to $\boldsymbol{x}$, i.e., the percentage of n-grams in $\boldsymbol{y}$ that are extracted from $\boldsymbol{x}$.[2] Following common practice (Papineni et al., 2002), we use n-grams up to length four. We do not include density in $\chi(\boldsymbol{x}, \boldsymbol{y})$ as its range is unbounded. The measure lcsr (longest common sub-

[2] We smooth all n-gram counts (Chen and Cherry, 2014) to avoid undefined or zero harmonic mean values in highly abstractive summaries. See Appendix A for details.

x the supreme court reserved its verdict on a batch of pleas which have raised questions

y the supreme court reserved its decision on a batch of pleas that have raised questions

Figure 3: Example of input and highly extractive generated output. The color coding is the same as in Fig. 1.

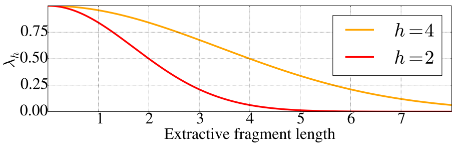


Figure 4: λ_h defines discounts for extractive fragments based on their lengths. Smaller h values lead to more abstractive summaries.

sequence ratio), short for $\text{lcsr}(\boldsymbol{x}, \boldsymbol{y})$, is the length of the longest common subsequence (LCS) between $\boldsymbol{x}$ and $\boldsymbol{y}$ divided by the length of $\boldsymbol{y}$. lcsr, inspired by ROUGE-L (Lin, 2004), generalizes perfect fusion$_k$ to consider *all* instances of non-contiguous overlaps between input and summary. Adding a measure of non-contiguous overlap is important as it detects overlaps that are long but broken up by minor changes, such as synonyms, as in the example in Figure 3. Finally, the MINT (**M**etric for lexical **in**dependence of generated **t**ext) abstractiveness measure is defined as $\text{MINT}(\boldsymbol{x}, \boldsymbol{y}) = 1 - \chi(\boldsymbol{x}, \boldsymbol{y})$. For a set of inputs and their summaries, we report the average MINT score. See Figure 1 for the MINT scores of three increasingly abstractive example summaries. In Section 5, we show that MINT scores correlate hightly with density scores.

The described MINT score capitalizes on prior work to provide a comprehensive and unified metric for abstractiveness of conditionally generated text, combining measures of contiguous and non-contiguous overlap into a single percentage score. The implementation of MINT we provide will facilitate standardized comparisons of abstractiveness across different works.

2.2 Nonlinear Abstractiveness Constraints

We now introduce nonlinear abstractiveness constraints (NAC), which enable us to control the degree of abstractiveness at decoding time; it will allow us to use a trained summarization model to decode input multiple times while applying constraints to control the abstractiveness of the generated text output (e.g., see Figure 1). We will use this technique to analyze the impact of abstractiveness on factuality (Section 4).

Let $\mathcal{F}(\boldsymbol{x}, \boldsymbol{y})$ be the set of the longest extractive fragments in the decoding output $\boldsymbol{y}$ with respect to the input $\boldsymbol{x}$. In Figure 1, such fragments are marked in color for each summary. We define a function $\lambda_h(|\boldsymbol{f}|)$ that assigns a discount probability to any extractive fragment $\boldsymbol{f} \in \mathcal{F}(\boldsymbol{x}, \boldsymbol{y})$:

$$\lambda_h(|\boldsymbol{f}|) = 2^{-|\boldsymbol{f}|^2/h^2} \tag{1}$$

We configure this function[3] with h, interpreted as the length of an extracted fragment for which $\lambda_h = 0.5$. Decreasing h results in a λ_h that discounts shorter extractive fragments more strongly, leading to increased abstractiveness (see Figure 4). Our discount penalty grows nonlinearly, affecting longer extractive fragments more strongly than multiple shorter ones with the same combined length. To see why we choose a **nonlinear penalty**, consider for example that extracting a 10-gram makes a summary more extractive than using ten words from the article separately, since an extracted 10-gram will be highly recognizable as stemming from the input. This nonlinearity is in contrast to Weber et al. (2018), which used a linear penalty to control the amount of copying in a pointer network.

In decoding, we search for the summary $\hat{\boldsymbol{y}}$ that maximizes the product of the summarization model probability, $p_\text{M}(\boldsymbol{y} \mid \boldsymbol{x})$, and the discount probabilities of the extractive fragments $\mathcal{F}(\boldsymbol{x}, \boldsymbol{y})$:

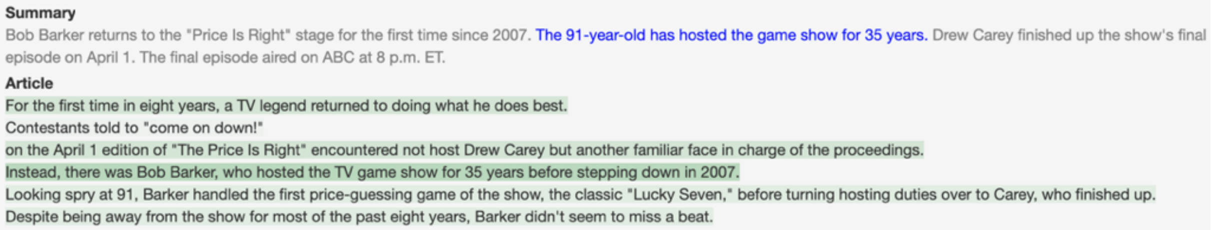
$$\hat{\boldsymbol{y}} = \arg\max_{\boldsymbol{y}} p_\text{M}(\boldsymbol{y} \mid \boldsymbol{x}) \times \prod_{f \in \mathcal{F}(\boldsymbol{x}, \boldsymbol{y})} \lambda_h(|\boldsymbol{f}|) \tag{2}$$

Beam Decoding. The model probability $p_\text{M}(\boldsymbol{x}, \boldsymbol{y})$ in neural text generation models (Section 5.1.1) decomposes for token-by-token decoding as $\prod_{i=1}^{|\boldsymbol{y}|} p_\text{M}(y_i \mid \boldsymbol{x}, y_1, \ldots, y_{i-1})$. Similarly, we decompose the application of the λ_h function for any partial or completed extractive fragment $\boldsymbol{f}$:

$$\lambda_h(|\boldsymbol{f}|) = \prod_{l=1}^{|\boldsymbol{f}|} \frac{\lambda_h(l)}{\lambda_h(l-1)} \tag{3}$$

Therefore, to successively apply λ_h at each output position i in beam decoding, each candidate for token y_i is evaluated to check whether choosing it would extend an extractive fragment to length l. If so, its model probability $p_\text{M}(y_i \mid \ldots)$ is multiplied with $\lambda_h(l)$ and the $\lambda_h(l-1)$ that was applied to the previous token y_{i-1} is divided out. We are not

[3] Additionally, the exponent used in $|\boldsymbol{f}|^2$ and h^2 could be configured, but we keep it at 2 in our experiments. A larger exponent would result in a steeper descent around h.



Summary

Bob Barker returns to the "Price Is Right" stage for the first time since 2007. The 91-year-old has hosted the game show for 35 years. Drew Carey finished up the show's final episode on April 1. The final episode aired on ABC at 8 p.m. ET.

Article

For the first time in eight years, a TV legend returned to doing what he does best.

Contestants told to "come on down!"

on the April 1 edition of "The Price Is Right" encountered not host Drew Carey but another familiar face in charge of the proceedings.

Instead, there was Bob Barker, who hosted the TV game show for 35 years before stepping down in 2007.

Looking spry at 91, Barker handled the first price-guessing game of the show, the classic "Lucky Seven," before turning hosting duties over to Carey, who finished up.

Despite being away from the show for most of the past eight years, Barker didn't seem to miss a beat.

Figure 5: Screenshot (part) of a Mechanical Turk task (HIT) to judge the factuality of a summary sentence (in blue) with respect to news articles. Darker green article sentences are more similar to the blue summary sentence. The full task showed sentences from two more articles in the same cluster; from the Multi-News test set.

aiming to control the length of the generated output; instead we penalize the model in proportion to the length of any phrases it would extract from the input and encourage it to use novel phrases instead.

Extraction Rewards. We can choose to apply an extraction *reward*, rather than a penalty, by using the inverse $1/\lambda_h$; smaller values of h then result in summaries that are more *extractive*.

3 Factuality

We now describe metrics for factuality, before we can describe the relationship between abstractiveness and factuality (Section 4). By factuality of a summary $\boldsymbol{y}$, we mean factual consistency with the input $\boldsymbol{x}$, rather than objective factuality or universal truth. Measuring factuality automatically is an active area of research (Gabriel et al., 2020). Factuality is most naturally measured by human annotators; we describe our setup for human factuality annotation first, then move to automatic metrics.

3.1 Human-annotated Factuality

We use Amazon's Mechanical Turk (AMT) to measure the factuality of automatically generated summaries with human annotators. These annotators are untrained, so we use multiple mitigation strategies to obtain high-quality judgements. We simplify the task: To avoid overwhelming annotators with long text, we select a single sentence per summary and ask the annotators if it is factually consistent with the shown article(s). The other sentences of the summary are given as well for context, shown in gray (see Figure 5). The article(s) are shortened to show a total of 9 sentences that were determined to be semantically most similar to the selected summary sentence;[4] the remaining article parts are replaced by "…". The summary sentence is selected at random in proportion to its length. For each summary, we get judgements only for the randomly selected sentence. Aggregated over a set of summaries, we measure the average chance of any randomly selected summary sentence to be factual. We have verified high correlation of these factuality rates with the factuality rates obtained through professional annotators who judged complete summaries with respect to the full articles (see Appendix C).

We provide detailed task instructions, including examples for intrinsic and extrinsic factual errors (Maynez et al., 2020). We require that potential annotators pass a custom qualification test of finding factuality errors. Only workers with at least 100 completed tasks on AMT with an acceptance rate of 95%+ may take the test; 15% of those pass, enabling them to work on our tasks. We use three annotators per task and use MACE (Hovy et al., 2013) to aggregate annotations and recover the most likely binary factuality judgement per summary. We add summaries for which we know the correct factuality annotation and repeatedly check the annotators' accuracy on those summaries while they are annotating; all answers from annotators who fall below a threshold are replaced by answers from additional annotators. Appendix C describes more details on our setup and fair compensation.

For any set of generated summaries, we create the AMT tasks, get an aggregate binary judgement per summary based on the multiple answers as described, and report the mean of all human binary summary factuality judgements; we call this score **FactH** (Table 1). We collect human factuality judgements for 10.2k BART summaries with varying degrees of abstractiveness, and for 4.2k summaries from five different summarization models.

Released Datasets. We release these human judgements as datasets called ConstraintsFact (Section 5.1) and ModelsFact (Section 5.2). Previous datasets with human factuality judgements

[4] We measure cosine similarity of sentence encodings computed by the Universal Sentence Encoder (Cer et al., 2018).

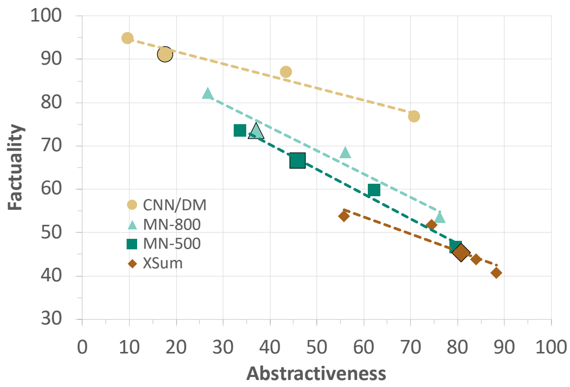


Figure 6: Human factuality judgements (FACTH) for different degrees of abstractiveness (MINT). Each color represents a BART model trained on a particular dataset, decoded with varying decoding constraints (Sec. 2.2); large outlined symbols mean no constraints.

(Wang et al., 2020; Kryscinski et al., 2020; Maynez et al., 2020; Pagnoni et al., 2021) are substantially smaller, with under 5k summaries each, and our CONSTRAINTSFACT dataset is the first that evaluates the factuality of summaries with systematically varied degrees of abstractiveness.

3.2 Automatically Measured Factuality

Measuring factuality *automatically* is an active research area; Pagnoni et al. (2021) gives an overview over recent metrics and compares their correlations to human judgements, where **DAE** (Goyal and Durrett, 2020, 2021) and **FactCC** (Kryscinski et al., 2020) perform well. DAE is an entailment model that classifies the factuality of the dependency arcs in the summary, resulting in fine-grained judgements at the subsentence level. FactCC is a BERT-based binary classifier trained on pairs of input and output sentences, where the output sentence is annotated as either factual or non-factual.

4 Abstractiveness-Factuality Tradeoff

The metrics for factuality and abstractiveness along with the abstractiveness constraints allow us to systematically explore the relationship between abstractiveness and factuality. We can control abstractiveness and observe the effect on factuality, i.e., we can vary the amount of lexical overlap between input and generated summary and observe the extent to which the summary preserves the input semantics.

Factuality Trend Lines. To explore this relationship, we train summarization models on different datasets. For any trained summarization model, we decode the test set multiple times with different h values for λ_h (Equation 1), resulting in sets of summaries with varying degrees abstractiveness. For each of these test set decodings, we measure abstractiveness using MINT and the corresponding factuality using human annotations, unless otherwise noted. This results in a series of (abstractiveness, factuality) points for any trained summarization model, which can be plotted, along with a linear trend line. Figure 6 shows such a plot; Section 5.1.2 discusses its details.

F@50 Score. Given each trend line, we can read off the factuality at 50% abstractiveness, an intuitively interpretable metric, which we call F@50; it provides a comparison of the factuality of different models with a fixed degree of abstractiveness.

MINT-adjusted Factuality Scores. We characterize the tradeoff on any single decoding output using a weighted average between factuality and abstractiveness, $(\phi F + A)/(\phi + 1)$. To measure abstractiveness A, we use MINT; to measure factuality F, we use human-measured factuality or an automatic metric with [0,1] range like DAE or FactCC, resulting in abstractiveness-adjusted factuality metrics μ**FactH**, μ**DAE**, μ**FactCC**, etc.

We give factuality a higher weight, since factual semantic representation of the input is a fundamental requirement for summarization and low factuality can have negative societal impact (Zellers et al., 2019), while abstractiveness is a desirable stylistic property. When two measures are combined into one comprehensive evaluation metric there is no *a priori* correct mixture weight; we follow common practice to give the more important measure twice the weight (Kohonen et al., 2010; Li et al., 2020; Preuß et al., 2021; Opitz and Frank, 2021) and set ϕ to 2. By this definition, a system whose factuality decreases by x units, as compared to another system, must make up for the lost factuality by $2x$ units in abstractiveness to get the same score. When two systems have the same factuality, the score prefers the one with higher abstractiveness.

4.1 Discussion

The abstractiveness-adjusted factuality metrics address the issue that in the past, factuality rates of different systems have been compared without taking abstractiveness into account. However, if one system has a higher factuality rate than another, it may

	λ	Mint	FactH	μFactH	F@50
CNN/DM	$1/\lambda_2$	9.7	94.8	66.5	84.4
	none	17.6	91.2	66.7	
	λ_4	43.5	87.0	72.5	
	λ_2	70.8	76.7	74.7	
MN-800	$1/\lambda_2$	26.8	82.2	63.7	68.9
	none	37.0	73.5	61.3	
	λ_4	56.1	68.5	64.4	
	λ_2	76.2	53.5	61.1	
MN-500	$1/\lambda_2$	33.6	73.5	60.2	64.4
	none	45.9	66.5	59.6	
	λ_4	62.3	59.7	60.6	
	λ_2	79.7	46.5	57.6	
XSum	$1/\lambda_1$	55.8	53.7	54.4	56.7
	$1/\lambda_2$	74.5	51.7	59.3	
	none	80.8	45.3	57.2	
	λ_4	84.0	43.7	57.1	
	λ_2	88.3	40.7	56.5	

Table 1: Abstractiveness and factuality on 600 test samples per setting. The 17 Mint and FactH numbers are as shown in Figure 6; we add μFactH and F@50.

have achieved this by copying phrases from the input into the summary with minimal rephrasing, i.e., by having a low degree of abstractiveness. Such a system may produce high-quality summaries, but their factuality rate cannot directly be compared to the factuality numbers of more abstractive summarization systems. Summarization methods that are highly factual and abstractive are able to rephrase the input with few factual errors; when we compare the factuality of abstractive summarizers we must control for the amount of such rephrasing. The abstractiveness-adjusted factuality metrics we propose enable us to compare the factuality of abstractive summarization models even when they perform different amounts of rephrasings.

As an analogy, consider precision and recall. High precision can be trivially achieved with low recall, just as high factuality can be achieved with low abstractiveness. Therefore when comparing the precision of different retrieval systems, their recall numbers are taken into account by using the F-score.[5] Similarly, we argue that factuality comparisons must take abstractiveness into account.

Dataset	Train	Valid	Test
CNN/DM	287,227	13,368	11,490
XSum	204,045	11,332	11,334
Multi-News	44,972	5,622	5,622

Table 2: Train/valid/test split on public datasets.

5 Experiments

5.1 Comparison Across Datasets Using NAC

Datasets. We use CNN/DM (Hermann et al., 2015), XSum (Narayan et al., 2018), and Multi-News (Fabbri et al., 2019), all of which contain English-only text. CNN/DM contains news articles from CNN and DailyMail paired with bullet point summaries. XSum contains articles from BBC News, using each article's first sentence as summary.[6] In Multi-News, each summary is written by a professional editor and paired with a cluster of news articles. For all three public datasets, we use the provided training/validation/test split. The sizes of the three datasets are listed in Table 2. From each of the three datasets, we use 600 samples to compare human and automatic factuality judgements.[7]

5.1.1 Setup

We use the Bart (Lewis et al., 2020) sequence-to-sequence model, which was pretrained on 160GB of text and gives competitive results on CNN/DM and XSum. Our models use the provided model checkpoints for the CNN/DM and the XSum datasets as well as the recommended decoding settings. For Multi-News (MN), we train a model on the training set, starting from the `bart.large` pretrained model.[8] For Multi-News, we truncate the input documents per cluster so that their combined length does not exceed N words, following Fabbri et al. (2019). We train models with $N = 800$ and $N = 500$, called MN-800 and MN-500, respectively. We measure the Mint scores for the reference summaries in these datasets; these can be compared to the Mint scores obtained in

[5] In our case, we use a weighted arithmetic mean instead because an F score would steeply decline to zero as abstractiveness goes to zero, which is undesirable for output whose factuality is high.

[6] Following Wang et al. (2020), we reinsert the first sentences whenever we measure factuality of XSum summaries on AMT or with automatic metrics.

[7] For Multi-News and XSum, we take the first 600 samples per test set. For CNN/DM, we take the first 300 and the last 300 test samples, from CNN and Daily Mail, respectively.

[8] We train for five epochs (learning rate: 2e-5) and limit output to 50 to 300 tokens.

decoding (Section 5.1.2). The test set references for MN-500 have a MINT score of 78.2%, compared to 72.8% for MN-800. MINT is higher for MN-500 since the shorter truncation removes article content that could otherwise overlap with the summaries. The MINT scores for the CNN/DM and XSum references are 59.6% and 87.8%, respectively; XSum is the most abstractive dataset.

5.1.2 Results

We use each of the four BART models to decode its respective test set multiple times, with varying abstractiveness constraints, resulting in 17 outputs. For each one, we obtain human factuality judgements on the corresponding 600 samples, resulting in 17 x 600 human factuality judgements – our CONSTRAINTSFACT dataset –, which we aggregate into 17 mean FACTH scores; we also compute the corresponding 17 MINT scores. Figure 6 plots the resulting abstractiveness and human-measured factuality for each of the four models, thereby providing a visual representation of the abstractiveness-factuality tradeoff for these models. Table 1 shows the same 17 MINT and FACTH values, along with μFACTH and F@50 scores.

The lower right of Figure 6 shows five lozenges (♦). The larger one represents the decoding with our **XSum**-trained model using default settings; the other four red points represent decodings under the same model, but with different abstractiveness constraints that result in more *extractive* ($1/\lambda_h$) or more *abstractive* (λ_h) summaries (Section 2.2). The five red points are associated with a dashed linear trend line. Compared to the other points in the figure, abstractiveness is high and factuality low – the model tends to paraphrase its input, often incorrectly. It took a strong extractive reward ($1/\lambda_1$), which we did not use for the models trained on other datasets, to bias this model toward lower abstractiveness and higher factuality.

For the **Multi-News** models, four decodings using MN-500 are shown as squares (■), decodings under MN-800 as triangles (▲). The MN-800 model is more factual across the abstractiveness spectrum. This can be explained by the fact that for MN-500, larger parts of the input are truncated (Section 5.1.1) that the untruncated reference summary in training may still refer to; the MN-500 model learns to hallucinate more.

The four decodings for **CNN/DM** are shown as bullets (●). Its model output without abstractiveness constraint (large bullet) is the most extractive; the extraction reward to its left (using $1/\lambda_2$) cannot make it much more extractive; however, there is room to the right, and the abstraction rewards (λ_4 and λ_2) move its abstractiveness far into the abstractiveness level of Multi-News and XSum.

F@50 Scores. One of the main takeaways of this study is that different systems can have different factuality rates at the same level of abstractiveness. Previous authors have observed that XSum summaries are highly abstractive and less factual, and that CNN/DM summaries are at the opposite side of that spectrum. We confirm this; however, we add that we can bias the XSum model to create less abstractive summaries and the CNN/DM model to create more abstractive models, so that **their abstractiveness becomes comparable**, and the factuality rates still differ considerably: Based on the trend line, the F@50 score of the XSum model is 56.7%, while the CNN/DM model's F@50 is 84.4%. MN-800 and MN-500 lie in the middle.

μFACTH Scores. The μFACTH scores adjust FACTH for abstractiveness. They penalize the CNN/DM model for its low abstractiveness and reward the XSum model for its high abstractiveness, bringing them closer together, compared to their more divergent FACTH scores. The μFACTH scores for MN-800 and MN-500 are also close (59.6% versus 61.3% for λ=none), as MN-800 is more factual but also less abstractive.

Summary Quality and Abstractiveness. Table 3 lists ROUGE-L scores for the different decodings, along with abstractiveness metrics, measured on the *full* test sets. ROUGE scores aim to measure summary quality by comparing the generated summaries with the reference summaries, while abstractiveness metrics measure overlap between the generated summaries and the input. Decodings without abstractiveness constraints replicate previous works' ROUGE scores (Lewis et al., 2020; Fabbri et al., 2019) (Appendix H). The λ_4 constraint can **dramatically increase abstractiveness while leaving ROUGE scores virtually unchanged**. We also conduct a human evaluation of informativeness and coherence, comparing unconstrained summaries with summaries generated with the λ_4 decoding constraint; the unconstrained decoding is preferred for XSum but the constrained decoding is preferred for CNN/DM, and results are mixed for Multi-News, see Appendix D. The density scores (Grusky et al., 2018) in the table have high correla-

	λ	RL	MINT	p3	p4	lcsr	density
CNN/DM	$1/\lambda_2$	37.9	9.0	89.0	84.7	93.1	28.9
	none	41.0	16.8	79.5	72.1	89.4	15.4
	λ_4	41.5	43.7	50.0	35.1	77.8	4.6
	λ_2	39.3	70.3	26.4	12.6	67.4	2.2
MN-800	$1/\lambda_2$	44.8	26.6	71.1	64.1	69.5	20.7
	none	45.8	37.1	58.9	50.1	63.3	13.4
	λ_4	45.8	56.3	38.7	27.0	51.9	4.3
	λ_2	44.0	76.4	20.7	10.4	41.6	2.0
MN-500	$1/\lambda_2$	44.6	34.1	63.7	56.4	61.0	17.6
	none	45.5	45.9	50.2	41.4	54.2	10.6
	λ_4	45.1	62.2	33.4	22.7	44.8	3.6
	λ_2	43.3	79.8	17.8	8.8	35.9	1.8
XSum	$1/\lambda_1$	30.8	53.8	41.7	32.3	66.9	5.8
	$1/\lambda_2$	36.0	73.9	23.0	14.1	57.7	3.0
	none	36.8	80.2	17.6	9.2	54.5	2.4
	λ_4	36.8	83.6	14.6	6.6	52.8	2.2
	λ_2	36.3	88.1	10.8	4.1	49.8	1.9

Table 3: Impact of λ on ROUGE-L F_1 (RL) and abstractiveness metrics on the full test sets. p3, p4, lcsr are component scores in MINT (Sec. 2.1), density is average length of extracted fragments (Grusky et al., 2018). ROUGE measures overlap with reference summaries, abstractiveness metrics measure input overlap.

	Model	MINT	μFACTH		μDAE		μFactCC	
CNN/DM	BART	16.8	**66.4**	91.2	**67.4**	92.6	56.2	75.9
	BERTSUM	14.1	64.7	90.0	57.8	79.6	57.0	78.5
	PGCONV	5.5	63.5	**92.5**	64.0	**93.3**	62.3	**90.7**
	BOTTOMUP	17.2	50.6	67.3	55.0	73.9	54.3	72.9
	ABSRL	**18.9**	60.6	81.5	62.3	84.0	**64.1**	86.8
XSum	BART	80.2	**56.9**	**45.3**	**67.3**	**60.8**	**53.9**	**40.8**
	BERTSUM	**82.8**	52.1	36.8	61.5	50.8	50.8	34.8

Table 4: Abstractiveness (MINT) and factuality of different models. For each factuality metric, we first list its MINT-adjusted variant in green. Example: BART's μFACTH is 66.4, while the unadjusted FACTH is 91.2. All numbers are percentage scores $\in$ [0,100].

tion with the MINT scores.

5.2 Comparison Across Different Models

We also compare the abstractiveness-factuality tradeoffs of summarization models from the literature. We obtain outputs of four summarization models other than **BART**: **BERTSUM** (Liu and Lapata, 2019) is a transformer model in which only the encoder is pretrained; **PGCONV** (See et al., 2017) is a pointer-generator network; **BOTTOMUP** (Gehrmann et al., 2018) and **ABSRL** (Chen and Bansal, 2018) select source fragments to constrain an abstractive generation model. We obtain human factuality judgements of the five model outputs on 600 samples of CNN/DM and XSum, respectively, and release this as our MODELSFACT dataset; we apply automatic metrics (e.g., DAE) as well as our abstractiveness-adjusted variants (e.g., μDAE) to the *full* test sets. Table 4 shows the results. For CNN/DM, we find that the highly extractive model PGCONV receives the highest automatic and human factuality scores, while the abstractiveness-adjusted variants favor BART or ABSRL, whose outputs represent better tradeoffs between abstractiveness and factuality. On **XSum**, BART's output is considerably more factual than BERTSUM's across all factuality metrics, while BART has only slightly lower abstractiveness; as a result, BART is also favored by all MINT-adjusted factuality metrics. Detailed results including additional factuality metrics are described in Appendix G.

The MINT-adjusted variants of factuality metrics put factuality rates into perspective. We encourage authors who compare factuality rates across summarization models to also compare MINT-adjusted variants (e.g., μDAE), to account for differing levels of abstractiveness.

6 Related Work

Abstractiveness-Factuality Tradeoff: Durmus et al. (2020) observe that abstractiveness at test time depends on the abstractiveness of the training data and that highly abstractive summaries tend to be less factual. We control for abstractiveness and see that factuality rates between different systems can vary widely at the *same* abstractiveness levels. Recently, Ladhak et al. (2022) present an alternative framework to evaluate the faithfulness-extractiveness tradeoff, requiring training multiple models on subsets of the training data to measure the tradeoff, while we use constraints to analyze tradeoffs that a single model makes. **Increasing Abstractiveness:** Kryściński et al. (2018) use policy gradient with a novelty reward to encourage abstraction in a pointer-generator (PG) (Gulcehre et al., 2016; See et al., 2017). Weber et al. (2018) penalize copying tokens during PG decoding. Our constraints apply to general sequence-to-sequence models and include nonlinear penalties. Song et al. (2020) control copying in training abstractive summarization models by masking the summary tokens with different probabilities, depending on whether they are seen in the input document or not. In contrast, our technique does not require retraining to obtain varying degrees of abstractiveness.

7 Conclusions

We presented new metrics and datasets for evaluating the relationship of abstractiveness and factuality. As part of our analysis, we presented abstractiveness constraints, which can bias a summarization model to increase or decrease the level of abstractiveness while generating summaries, using nonlinear penalties or rewards based on the length of summary fragments extracted from the source. Through automatic and human factuality evaluations, including 10.2k human factuality judgements of summaries with systematically varied abstractiveness, we shed light on how abstractiveness interacts with factuality, across multiple datasets and models. We proposed new metrics to measure the tradeoff, including F@50 and MINT-adjusted factuality rates, such as μDAE and μFactCC, and we established baselines for future research.

Limitations

The abstractiveness constraints we have presented can be used to increase or decrease the abstractiveness of the generated text. Dedicated code is needed to integrate such constraints into a decoder. The constraints are needed to obtain trend lines as in Figure 6, as well as the F@50 score. However, the MINT-adjusted factuality scores, such as μFactH, μDAE or μFactCC can be computed for any summarization system, without the need for implementing abstractiveness constraints, as we have done in Section 5.2.

Ethical Considerations

We have analyzed the factuality of generated text in relation to the abstractiveness of the source texts; we have also proposed new metrics that let researchers compare the factuality of different generative models. As such, we consider our work a contribution toward text generation methods that make fewer factual mistakes and become therefore more reliable and responsible. However, any advance in text generation methods can be used by bad actors to cheaply generate misleading or harmful texts.

We hired annotators on the Mechanical Turk platform to judge machine-generated summaries. Our first ethical consideration with respect to this data collection is fair and prompt pay for the work of the annotators. We describe in Appendix C that we paid all human subjects a fair average pay of \$12.50 USD per hour, based on observed median time spent per HIT. As described (Section 3.1), we automatically approved the annotators' work promptly and paid bonuses as appropriate. The annotators' privacy and confidentiality were respected at all times.

References

Shuyang Cao and Lu Wang. 2021. CLIFF: Contrastive learning for improving faithfulness and factuality in abstractive summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6633–6649, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2017. Faithful to the original: Fact aware neural abstractive summarization. *CoRR*, abs/1711.04434.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal Sentence Encoder. *EMNLP 2018 - Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Proceedings*, pages 169–174.

Boxing Chen and Colin Cherry. 2014. A systematic comparison of smoothing techniques for sentence-level BLEU. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 362–367, Baltimore, Maryland, USA. Association for Computational Linguistics.

Yen-Chun Chen and Mohit Bansal. 2018. Fast Abstractive Summarization with Reinforce-Selected Sentence Rewriting. In *Proc. of ACL*.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.

Esin Durmus, He He, and Mona Diab. 2020. Feqa: A question answering evaluation framework for faithfulness assessment in abstractive summarization. In *Association for Computational Linguistics (ACL)*.

H. P. Edmundson. 1969. New methods in automatic extracting. *J. ACM*, 16:264–285.

Günes Erkan and Dragomir R. Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Int. Res.*, 22(1):457–479.

Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1074–1084, Florence, Italy. Association for Computational Linguistics.

Tobias Falke, Leonardo F.R. Ribeiro, Prasetya Ajie Utama, Ido Dagan, and Iryna Gurevych. 2020. Ranking generated summaries by correctness: An interesting but challenging application for natural language inference. In *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, pages 2214–2220. Association for Computational Linguistics (ACL).

Lisa Fan, Dong Yu, and Lu Wang. 2018. Robust Neural Abstractive Summarization Systems and Evaluation against Adversarial Information. In *NIPS Interpretability and Robustness for Audio, Speech and Language Workshop*.

Saadia Gabriel, Asli Celikyilmaz, Rahul Jha, Yejin Choi, and Jianfeng Gao. 2020. Go Figure! A Meta Evaluation of Factuality in Summarization. Technical report.

Kavita A. Ganesan, ChengXiang Zhai, and Jiawei Han. 2010. Opinosis: A graph based approach to abstractive summarization of highly redundant opinions. In *International Conference on Computational Linguistics*.

Shen Gao, Xiuying Chen, Piji Li, Zhangming Chan, Dongyan Zhao, and Rui Yan. 2019. How to write summaries with patterns? learning towards abstractive summarization through prototype editing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3741–3751, Hong Kong, China. Association for Computational Linguistics.

Sebastian Gehrmann, Yuntian Deng, and Alexander M. Rush. 2018. Bottom-Up Abstractive Summarization. In *Proc. of EMNLP*.

Sebastian Gehrmann, Zachary Ziegler, and Alexander Rush. 2019. Generating abstractive summaries with finetuned language models. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 516–522, Tokyo, Japan. Association for Computational Linguistics.

Pierre-Etienne Genest and Guy Lapalme. 2012. Fully abstractive approach to guided summarization. In *Annual Meeting of the Association for Computational Linguistics*.

Ben Goodrich, Vinay Rao, Peter J Liu Mohammad Saleh, Google Brain, Peter J Liu, and Mohammad Saleh. 2019. Assessing The Factual Accuracy of Generated Text. In *International Conference on Knowledge Discovery and Data Mining (KDD)*.

Tanya Goyal and Greg Durrett. 2020. Evaluating factuality in generation with dependency-level entailment. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3592–3603, Online. Association for Computational Linguistics.

Tanya Goyal and Greg Durrett. 2021. Annotating and modeling fine-grained factuality in summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1449–1462, Online. Association for Computational Linguistics.

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

Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the unknown words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 140–149.

Karl Moritz Hermann, Tomáš Kočiskỳ, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 1*, pages 1693–1701.

Dirk Hovy, Taylor Berg-Kirkpatrick, Ashish Vaswani, and Eduard Hovy. 2013. Learning whom to trust with MACE. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1120–1130, Atlanta, Georgia. Association for Computational Linguistics.

Oskar Kohonen, Sami Virpioja, and Krista Lagus. 2010. Semi-supervised learning of concatenative morphology. In *Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology*, pages 78–86, Uppsala, Sweden. Association for Computational Linguistics.

Wojciech Kryscinski, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Neural text summarization: A critical evaluation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 540–551, Hong Kong, China. Association for Computational Linguistics.

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346.

Wojciech Kryściński, Romain Paulus, Caiming Xiong, and Richard Socher. 2018. Improving abstraction in text summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1808–1817, Brussels, Belgium. Association for Computational Linguistics.

Faisal Ladhak, Esin Durmus, He He, Claire Cardie, and Kathleen McKeown. 2022. Faithful or extractive? on mitigating the faithfulness-abstractiveness trade-off in abstractive summarization. In *Proc. of ACL*.

Logan Lebanoff, John Muchovej, Franck Dernoncourt, Doo Soon Kim, Seokhwan Kim, Walter Chang, and Fei Liu. 2019. Analyzing sentence fusion in abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 104–110, Hong Kong, China. Association for Computational Linguistics.

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

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

Xiangci Li, Hairong Liu, and Liang Huang. 2020. Context-aware stand-alone neural spelling correction. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 407–414, Online. Association for Computational Linguistics.

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

Yang Liu and Mirella Lapata. 2019. Text Summarization with Pretrained Encoders. In *Proc. of EMNLP*.

Hans Peter Luhn. 1958. The automatic creation of literature abstracts. *IBM J. Res. Dev.*, 2:159–165.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

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

Joel Larocca Neto, Alex Alves Freitas, and Celso A. A. Kaestner. 2002. Automatic text summarization using a machine learning approach. In *Proceedings of the 16th Brazilian Symposium on Artificial Intelligence: Advances in Artificial Intelligence*, SBIA '02, page 205–215, Berlin, Heidelberg. Springer-Verlag.

Juri Opitz and Anette Frank. 2021. Towards a decomposable metric for explainable evaluation of text generation from AMR. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1504–1518, Online. Association for Computational Linguistics.

Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. 2021. Understanding factuality in abstractive summarization with FRANK: A benchmark for factuality metrics. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4812–4829, Online. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Thierry Poibeau and Horacio Saggion. 2012. Automatic Text Summarization: Past, Present and Future. In *Multi-source, Multilingual Information Extraction and Summarization*, pages 3–13.

Svenja Preuß, Luna Pia Bley, Tabea Bayha, Vivien Dehne, Alessa Jordan, Sophie Reimann, Fina Roberto, Josephine Romy Zahm, Hanna Siewerts, Dirk Labudde, and Michael Spranger. 2021. Automatically identifying online grooming chats using CNN-based feature extraction. In *Proceedings of the 17th Conference on Natural Language Processing (KONVENS 2021)*, pages 137–146, Düsseldorf, Germany. KONVENS 2021 Organizers.

Dragomir R. Radev and Kathleen R. McKeown. 1998. Generating natural language summaries from multiple on-line sources. *Computational Linguistics*, 24(3):469–500.

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

Leonardo F. R. Ribeiro, Mengwen Liu, Iryna Gurevych, Markus Dreyer, and Mohit Bansal. 2022. FactGraph: Evaluating factuality in summarization with semantic graph representations. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3238–3253, Seattle, United States. Association for Computational Linguistics.

Horacio Saggion and Guy Lapalme. 2002. Generating indicative-informative summaries with sumum. *Computational Linguistics*, 28:497–526.

Thomas Scialom, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. 2019. Answers unite! unsupervised metrics for reinforced summarization models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3246–3256, Hong Kong, China. Association for Computational Linguistics.

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

Kaiqiang Song, Bingqing Wang, Zhe Feng, Ren Liu, and Fei Liu. 2020. Controlling the amount of verbatim copying in abstractive summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34(05), pages 8902–8909.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and answering questions to evaluate the factual consistency of summaries. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5008–5020.

Noah Weber, Leena Shekhar, Niranjan Balasubramanian, and Kyunghyun Cho. 2018. Controlling decoding for more abstractive summaries with copy-based networks. *arXiv preprint arXiv:1803.07038*.

Kam-Fai Wong, Mingli Wu, and Wenjie Li. 2008. Extractive summarization using supervised and semi-supervised learning. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 985–992, Manchester, UK. Coling 2008 Organizing Committee.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, F. Roesner, and Yejin Choi. 2019. Defending against neural fake news. In *NeurIPS*.

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

Yuhao Zhang, Derek Merck, Emily Bao Tsai, Christopher D. Manning, and Curtis P. Langlotz. 2019. Optimizing the Factual Correctness of a Summary: A Study of Summarizing Radiology Reports.

A Measuring Abstractiveness with MINT

N-gram Overlap. Each p_n, short for $p_n(\boldsymbol{x}, \boldsymbol{y})$, is the n-gram precision of the n-grams in $\boldsymbol{y}$ with respect to $\boldsymbol{x}$, i.e., the percentage of n-grams in $\boldsymbol{y}$ that are extracted from $\boldsymbol{x}$.[9] For highly abstractive outputs, higher-order n-gram precision can be zero, leading to an undefined or zero harmonic mean value. We prevent this by smoothing the n-gram counts from which n-gram precisions are calculated, such that each n-gram count is the average of itself and the smoothed $(n-1)$-gram count and the unsmoothed $(n+1)$-gram count. The smoothed 0-gram count is defined as the 1-gram count plus one. We chose this method for its simplicity and effectiveness; it is described as method 5 in Chen and Cherry (2014).

Harmonic Mean. We use the harmonic mean, in analogy to the definition of the F_1 score, as it is a mean function designed to aggregate ratios with different denominators.

For a completely extractive summary that extracts sentences in the original order, the MINT score is 0. The score increases as the order of the extractive fragments is changed with respect to the input, their lengths are decreased and new words and fragments are introduced that are not part of the input $\boldsymbol{x}$. The use of the length-normalized LCS score (lcsr) is inspired by ROUGE-L; it is a useful addition to the n-gram precisions as it can detect the extraction of longer n-grams broken up by minor edits. As an example, consider the $(\boldsymbol{x}, \boldsymbol{y})$ pair shown in Figure 3. Only 4 of the 12 summary four-grams match the input, i.e., p_4=33.3%, although very high overlap is apparent due to the fact that a 15-word fragment from the input was extracted with only the words “verdict” and “which” minimally changed by synonym substitution. The lcsr score reflects this and measures 12/15=80.0% overlap. On the other hand, the n-gram precisions used in the MINT score are valuable in detecting textual overlaps that are not part of the longest common subsequence.

B Details on the Abstractiveness Constraints

Log Space. We have described the abstractiveness constraints in probability space. In practice, we equivalently search for $\hat{\boldsymbol{y}}$ in log space using log probabilities and the log of λ_h defined in Equation 1. It can be shown that $\log \lambda_h(|\boldsymbol{f}|) = \frac{-|\boldsymbol{f}|^2}{(1.20112 \times h)^2}$.

C Details on Our Mechanical Turk Setup

We provide additional details on the strategies we use to obtain high-quality judgements on Amazon Mechanical Turk. We give detailed instructions to the annotators, with definitions and examples of different factual errors (see Figure 7). We also add a request to write a short explanation when a sentence is judged as not factual.

Tasks with Known Answers. We add a number of tasks with known answers, enabling us to estimate the accuracy of workers who work on multiple of these.

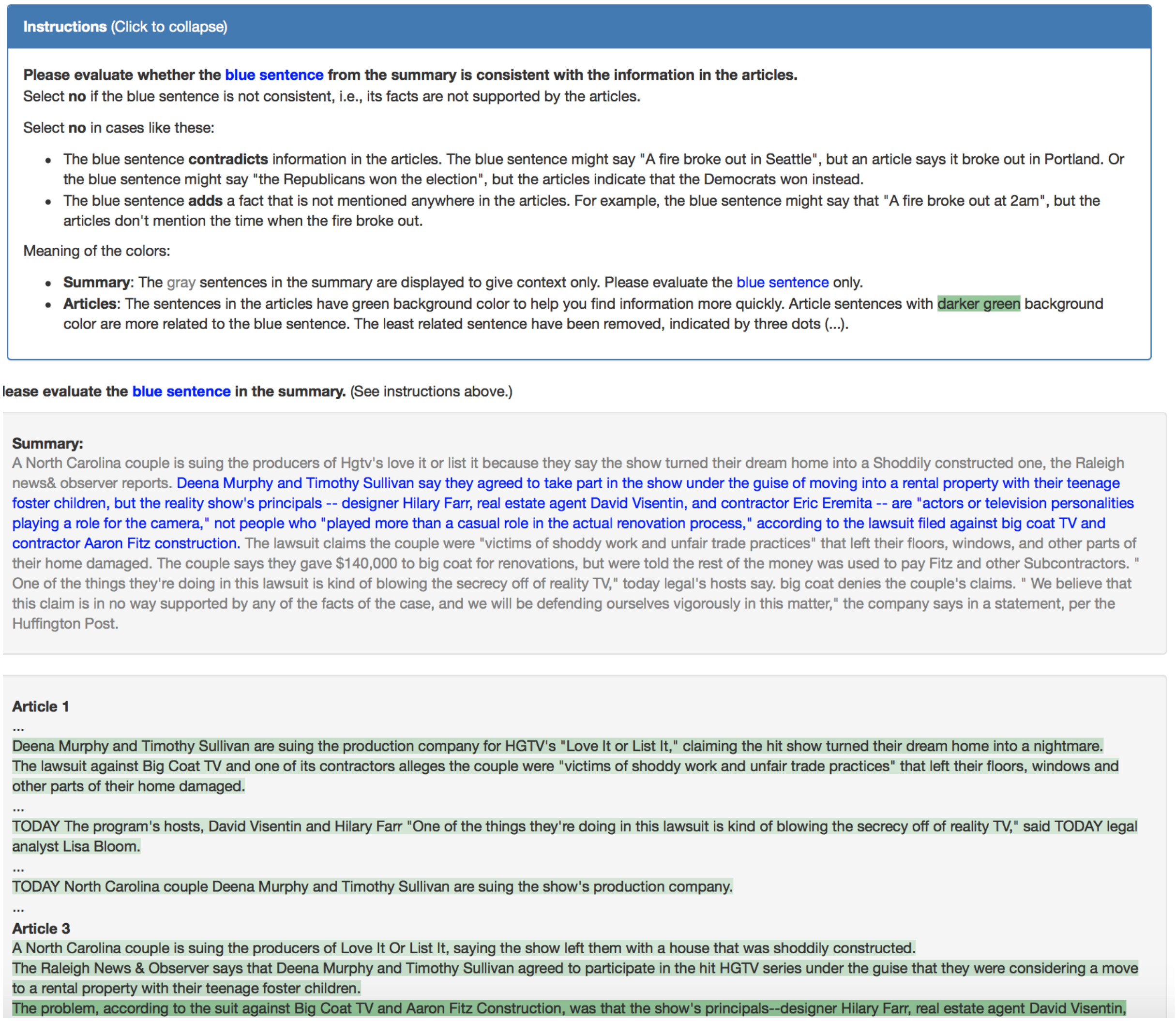
Automatic Quality Checks. Workers who complete the tasks too quickly, write no or very short explanation texts or have low accuracy on the tasks with known answers are automatically removed from our worker pool. Their answers are replaced with new answers.

Bonus. We use a bonus incentive structure. Every worker who passes the automatic quality checks receives a bonus at the end.

Check Against Professional Annotators. We have seven sets of 150 automatically generated summaries each, which we had previously sent to professional news editors to annotate factuality. Those annotators rated the complete summaries with respect to the complete inputs – no sentences were preselected to simplify the task. We re-annotated these summary-article pairs using our Mechanical Turk setup, and the resulting per-set factuality rates correlated highly (r=.88) with those previously obtained from the professional annotators ($p < .05$).

As a further quality check, we sent one set of 600 summaries to Mechanical Turk twice, several weeks apart. The two factuality rates obtained for that same set were close – 91.2% and 92.0%.

[9] MINT has elements of ROUGE (Lin, 2004) and BLEU (Papineni et al., 2002). We do not use the *modified* n-gram precisions, like BLEU does, because n-grams extracted multiple times from x should count as such every time.



Instructions (Click to collapse)

Please evaluate whether the blue sentence from the summary is consistent with the information in the articles.
Select **no** if the blue sentence is not consistent, i.e., its facts are not supported by the articles.

Select **no** in cases like these:

- The blue sentence **contradicts** information in the articles. The blue sentence might say "A fire broke out in Seattle", but an article says it broke out in Portland. Or the blue sentence might say "the Republicans won the election", but the articles indicate that the Democrats won instead.
- The blue sentence **adds** a fact that is not mentioned anywhere in the articles. For example, the blue sentence might say that "A fire broke out at 2am", but the articles don't mention the time when the fire broke out.

Meaning of the colors:

- **Summary**: The gray sentences in the summary are displayed to give context only. Please evaluate the blue sentence only.
- **Articles**: The sentences in the articles have green background color to help you find information more quickly. Article sentences with darker green background color are more related to the blue sentence. The least related sentence have been removed, indicated by three dots (...).

lease evaluate the blue sentence in the summary. (See instructions above.)

Summary:
A North Carolina couple is suing the producers of Hgtv's love it or list it because they say the show turned their dream home into a Shoddily constructed one, the Raleigh news& observer reports. Deena Murphy and Timothy Sullivan say they agreed to take part in the show under the guise of moving into a rental property with their teenage foster children, but the reality show's principals -- designer Hilary Farr, real estate agent David Visentin, and contractor Eric Eremita -- are "actors or television personalities playing a role for the camera," not people who "played more than a casual role in the actual renovation process," according to the lawsuit filed against big coat TV and contractor Aaron Fitz construction. The lawsuit claims the couple were "victims of shoddy work and unfair trade practices" that left their floors, windows, and other parts of their home damaged. The couple says they gave $140,000 to big coat for renovations, but were told the rest of the money was used to pay Fitz and other Subcontractors. " One of the things they're doing in this lawsuit is kind of blowing the secrecy off of reality TV," today legal's hosts say. big coat denies the couple's claims. " We believe that this claim is in no way supported by any of the facts of the case, and we will be defending ourselves vigorously in this matter," the company says in a statement, per the Huffington Post.

Article 1
...
Deena Murphy and Timothy Sullivan are suing the production company for HGTV's "Love It or List It," claiming the hit show turned their dream home into a nightmare.
The lawsuit against Big Coat TV and one of its contractors alleges the couple were "victims of shoddy work and unfair trade practices" that left their floors, windows and other parts of their home damaged.
...
TODAY The program's hosts, David Visentin and Hilary Farr "One of the things they're doing in this lawsuit is kind of blowing the secrecy off of reality TV," said TODAY legal analyst Lisa Bloom.
...
TODAY North Carolina couple Deena Murphy and Timothy Sullivan are suing the show's production company.
...
Article 3
A North Carolina couple is suing the producers of Love It Or List It, saying the show left them with a house that was shoddily constructed.
The Raleigh News & Observer says that Deena Murphy and Timothy Sullivan agreed to participate in the hit HGTV series under the guise that they were considering a move to a rental property with their teenage foster children.
The problem, according to the suit against Big Coat TV and Aaron Fitz Construction, was that the show's principals--designer Hilary Farr, real estate agent David Visentin,

Figure 7: Instructions for the factuality annotation task on Amazon Mechanical Turk, as well as the summary and part of the article text shown to the worker.

Qualification Test. For all our evaluations on Mechanical Turk (see Section 3.1), we first set up a short qualification test that can be taken by any worker from a country whose main language is English, who has completed 100 or more HITs so far with an acceptance rate of 95% or higher. The qualification test consists of just three questions from our factual consistency setup; two of which must be answered correctly, along with an explanation text (5 words or more) to explain when "not factually consistent" was chosen. 53% of workers who start the test provide answers to all three questions, and 27.6% of these answer at least two correctly and provide a reasonable explanation text, i.e., only 14.6% of the test takers are granted the qualification.

The qualification enables workers to work on our factual consistency HITs as well as our HITs judging informativeness and coherence.

Fair Compensation. The factual consistency task pays $0.15 per HIT with a bonus of $0.05. It can be done quickly, given the fact that a single summary sentence is evaluated and the related sentences in the article are highlighted. The task of evaluating informativeness and coherence (see Appendix D) pays $0.50 per HIT with a bonus of $0.25, as more text is displayed, compared to the factuality task. These amount to an average pay of $12.50 per hour, including the bonus, based on median time spent per HIT. The bonus is paid to workers who spend at least 10 seconds per HIT, give short explanation texts for their decisions and maintain high accuracy on HITs with known answers.

	CNN/DM		MN-800		XSum	
	inf.	coh.	inf.	coh.	inf.	coh.
prefer off	36.5	36.7	39.8	35.8	18.8	18.7
prefer λ_4	46.5	39.2	34.7	39.8	16.5	16.3
both equal	17.0	24.2	25.5	24.3	64.7	65.0

Table 5: Human quality evaluation of summaries generated with no abstractiveness constraint ("off") versus λ_4. We asked which summary is more informative or coherent, respectively. MN-800 stands for Multi-News with the input documents truncated to 800 words total (Section 5.1.1).

D Human Evaluation of Informativeness and Coherence

We conduct a human evaluation to determine the informativeness and coherence of the summaries generated with the λ_4 decoding constraint (Equation 1), which increases abstractiveness, as compared to not using any abstractiveness constraint. We use the same setup as for the factuality task, including a qualification test, three annotators per task and aggregation using MACE.

We use the following definitions of *informativeness* and *coherence* for the human evaluation:

- *Informativeness*: The more informative summary is better at expressing the main points of the news story. It contains information that is more relevant and important. It has fewer unimportant details. Its content is more similar to the human-written summary.

- *Coherence*: The more coherent summary has better structure and flow, is easier to follow. The facts are presented in a more logical order.

The results are shown in Table 5. For the **CNN/DM** model, the output without decoding constraints is the most extractive, and the raters preferred the more abstractive version generated with the decoding constraint, both for informativeness and coherence. For the **XSum** model, where the output with the decoding constraint disabled is already highly abstractive, the result is reversed. For **Multi-News**, the result is mixed: Raters found the output with no decoding constraints more informative, but less coherent.

Data	Size	DAE	FactCC	FEQA	QAGS
All	4.2k	.44	.35	.27	.44
CNN/DM	3.0k	.35	.24	.05	.27
XSum	1.2k	.39	.17	†.01	.25

Table 6: Pearson correlations to human factuality judgements on the ModelsFact dataset. The result with the † symbol is not significant.

E More On Automatic Factuality Metrics

When we apply FactCC to a summary, we apply it separately to each summary sentence and use the mean score per summary. For each sentence that we score with FactCC, we shorten the input document by selecting ten sentences with the highest cosine embedding similarity (Conneau et al., 2017), in order to fit the input to the length limits.

In the following two appendix sections, we use not only DAE and FactCC, as described in the main text, but also two metrics based on question answering: FEQA (Durmus et al., 2020) and QAGS (Wang et al., 2020). **FEQA** generates questions from masked summary sentences whose masked entities are used as "gold" answers; these are compared to the answers obtained from a QA model on the input. In **QAGS**, a question generation model generates questions from the summary, a QA model answers these questions from both summary and input, and the similarity of the answer pairs is evaluated.

F Correlating Human and Automatic Factuality Judgements

Table 6 shows correlations of the human judgements with different automatic metrics on the ModelsFact dataset, complementing earlier studies (Gabriel et al., 2020; Pagnoni et al., 2021). We compute correlations at the level of individual summaries. To make meaningful comparisons between the human and the automatic scores, we apply the automatic metrics here to the *single* randomly selected sentence per summary that the human annotators judged. Overall, we observe here that DAE has the highest correlations with human judgements.

Data	Model	MINT	μFACTH		μDAE		μFactCC		μFEQA		μQAGS	
CNN/DM	BART	16.8	**66.4**	91.2	**67.4**	92.6	56.2	75.9	47.2	62.4	**61.7**	84.2
	BERTSUM	14.1	64.7	90.0	57.8	79.6	57.0	78.5	47.6	64.4	60.8	84.2
	PGCONV	5.5	63.5	**92.5**	64.0	**93.3**	62.3	**90.7**	45.2	**65.0**	58.1	**84.4**
	BOTTOMUP	17.2	50.6	67.3	55.0	73.9	54.3	72.9	47.3	62.3	58.2	78.7
	ABSRL	**18.9**	60.6	81.5	62.3	84.0	**64.1**	86.8	**49.6**	65.0	61.3	82.5
XSum	BART	80.2	**56.9**	**45.3**	**67.3**	**60.8**	**53.9**	**40.8**	**50.9**	**36.2**	**53.4**	**40.1**
	BERTSUM	**82.8**	52.1	36.8	61.5	50.8	50.8	34.8	46.6	28.4	46.0	27.6

Table 7: Abstractiveness (MINT) and factuality of different summarization models. For each factuality metric, we first list its MINT-adjusted variant in green. Example: BART's μFACTH is 66.4, while the unadjusted FACTH is 91.2. All numbers are percentage scores ∈ [0,100].

G Comparison Across Different Models

Here we offer an extended description of our comparison of the abstractiveness-factuality tradeoffs of summarization models from the literature, including the use of additional automatic factuality metrics (see Appendix E).

Table 7 shows human and automatic factuality scores, as well as MINT-adjusted versions of these scores. We observe that all factuality metrics favor the output of the PGCONV model on **CNN/DM**; however, its low abstractiveness indicates that its output falls into the "trivially factual" quadrant (Figure 2). The MINT-adjusted variants (shown in green) penalize such low abstractiveness, favoring the BART or ABSRL models instead, whose outputs represent better tradeoffs between abstractiveness and factuality. Human factuality raters (FACTH) rank ABSRL in fourth place, while FactCC, FEQA and QAGS rank it highly; we hypothesize that ABSRL makes factual errors that these measures cannot detect well. On **XSum**, BART's output is considerably more factual than BERTSUM's across all factuality metrics, while BART has only slightly lower abstractiveness; as a result, BART is also favored by all MINT-adjusted factuality metrics. BART's pretraining of both encoder and decoder may be contributing to its factuality, in accordance with Maynez et al. (2020). Note that for DAE, we apply the Ent-C model on CNN/DM output and the XSUM-HUMAN model on XSum output. Appendix H.2 shows **ROUGE** scores.

H ROUGE Scores

H.1 BART Models

The aim of this paper is not to improve ROUGE scores, but to gain insights about the tradeoff between abstractiveness and factuality. We do, however, stress that the BART models we use in our analysis are competitive with the start of the art. We list our ROUGE-1, ROUGE-2 and ROUGE-L F_1 scores, as well as their averages; see the RL scores in Table 3 as well:

- For CNN/DM, our λ=none decoding has 44.1/21.2/41.0 with an average of 35.4, same as the average of 35.4 in Lewis et al. (2020).
- For XSum, our λ=none decoding has 45.3/21.9/36.8 with an average of 34.7, compared to an average of 34.9 in Lewis et al. (2020).
- For Multi-News, our MN-800 λ=none decoding has 50.2/20.5/45.8 with an average of 38.8, compared to improved ROUGE F_1 results of 44.5/16.0/40.3 with an average of 33.6 by Fabbri (personal communication) for Fabbri et al. (2019).

H.2 Comparing Summarization Models

To complement our comparison of different models in Section 5.2, we list the ROUGE-L F_1 scores of the five models in Table 8.

I Additional Experimental Details

We used AWS p3.8x and p3.16x EC2 machines for all our experiments, except we ran FEQA on the Multi-News summaries on a p3dn.24xlarge machine, as it required more memory.

	Model	RL
CNN/DM	BART	41.0
	BERTSUM	39.2
	PGCONV	36.4
	BOTTOMUP	38.3
	ABSRL	37.3
XSum	BART	36.8
	BERTSUM	31.3

Table 8: ROUGE-L F_1 scores for the models compared in Section 5.2.

The BART model has 406,290,432 parameters. Fine-tuning BART on the Multi-News training set took about 2.5 hours on 4 GPUs; we fine-tuned for 5 epochs following instructions on the fairseq BART webpage, without further hyperparameter search. For CNN/DM and XSum we used the provided checkpoints.[10] The minimum and maximum length for Multi-News decoding was determined by the lengths of the training reference summaries.

[10] See https://github.com/pytorch/fairseq/tree/master/examples/bart.