Don't Say What You Don't Know: Improving the Consistency of Abstractive Summarization by Constraining Beam Search

Daniel King[*] **Zejiang Shen**[*†] **Nishant Subramani**[†,♢] **Daniel S. Weld**[†,♡]
Iz Beltagy[†] **Doug Downey**[†,♣]

[†]Allen Institute for AI [♢]Masakhane [♡]University of Washington [♣]Northwestern University

daking@hey.com {shannons, nishants, danw, beltagy, dougd}@allenai.org

Abstract

Abstractive summarization systems today produce fluent and relevant output, but often "hallucinate" statements not supported by the source text. We analyze the connection between hallucinations and training data, and find evidence that models hallucinate because they train on target summaries that are unsupported by the source. Based on our findings, we present PINOCCHIO, a new decoding method that improves the consistency of a transformer-based abstractive summarizer by constraining beam search to avoid hallucinations. Given the model states and outputs at a given step, PINOCCHIO detects likely model hallucinations based on various measures of attribution to the source text. PINOCCHIO backtracks to find more consistent output, and can opt to produce no summary at all when no consistent generation can be found. In experiments, we find that PINOCCHIO improves the consistency of generation (in terms of F1) by an average of 67% on two abstractive summarization datasets.

1 Introduction

Abstractive text generation is an important task with the promise of compressing lengthy source material into concise summaries, satisfying application or user needs. Pretrained abstractive summarizers (e.g. BART (Lewis et al., 2020)) have recently achieved new state-of-the-art (SOTA) across multiple datasets (Fabbri et al., 2020). However, these systems remain unusable in most real world scenarios, because they frequently hallucinate information that is inconsistent with the input (Maynez et al., 2020).

Many researchers have proposed methods to assess and improve the consistency[1] of summarization systems. Two popular approaches are 1) incorporating extracted knowledge (Zhu et al., 2021) (possibly in the form of questions (Durmus et al., 2020)), and 2) incorporating a consistency text classifier (Kryscinski et al., 2020) (often based on natural language inference (NLI) (Falke et al., 2019)). These methods tend to reduce the problem of generating consistent text to another difficult problem (e.g. information extraction (IE) or NLI). Given a strong IE system or a structured representation of the source information, it is possible to dramatically improve the consistency of generated text (Zhang et al., 2020b; Tian et al., 2019), but such resources are only available in a narrow subset of domains.

Method	Text
Source	...The PSNI said the tablets were "as yet unidentified" but warned of the "potential dangers" they posed...
BART	A 17-year-old boy has been charged after a teenager was taken ill after taking what police have described as *"potentially lethal" ecstasy tablets*.
PINOCCHIO	A 17-year-old teenager has been charged with drugs offences after a teenager was treated in hospital after taking what police described as an "unidentified" drug.

Table 1: An example of hallucination. Inconsistent words are highlighted in *red italic* fonts. In this case, PINOCCHIO corrects the inconsistent detail in the BART output.

We propose a different approach for generating more consistent summaries. It is based on the observation that today's abstractive summarizers are often trained on gold summaries that contain statements unsupported by the source text (Matsumaru et al., 2020). This disconnect arises because the training datasets are built using distant supervision in order to scale, e.g. treating a news headline as a summary of its article or an encyclopedia entry as a summary of a portion of its references. We conjecture that a model optimized for likelihood and trained on target summaries containing unsupported statements will have a strong tendency to

[*]Denotes equal contribution.

[1]We use the terms "consistent" and "hallucinated" as antonyms, and avoid "factual". Check Section 2 for details.

hallucinate information rather than say something less "likely," but supported (§3). Further, common automatic evaluation metrics like ROUGE *reward* lexical similarity significantly more than consistency, preferring hallucinated lexically similar summaries to completely consistent lexically different ones.

Our method, called PINOCCHIO, is a novel decoding algorithm that constrains beam search to only consider predicted tokens that are likely to be supported by the source text. PINOCCHIO estimates which tokens are likely supported using simple but effective heuristics based on the model's confidence and attention distribution, and word frequency. When PINOCCHIO reaches a state where no supported token can be generated, it backtracks the search. It can also opt-out from generating a summary at all, rather than produce one expected to be hallucinated. We show how PINOCCHIO significantly improves consistency on two abstractive summarization datasets with only a small decrease in fluency, measured using careful human evaluations.

To test PINOCCHIO on diverse domains, we also develop a new abstractive summarization dataset called Scientific Concept Description (SCD). Inspired by the WikiSum (Liu* et al., 2018) dataset, SCD uses Wikipedia descriptions as the target summaries and the referenced papers as the source documents, detailed in (§5). It comes with a total of 60k samples of scientific concepts and 118k corresponding paper identifiers, with full text for 8k of the papers.

We make the following contributions:

1. We analyze the relationship between hallucination and training on targets that are not fully supported by the source.
2. We introduce PINOCCHIO, a decoding algorithm that improves generation consistency by constraining beam search to focus on input-supported tokens. It improves consistency by an average of 67% in two abstractive summarization datasets at the expense of a minor decrease to fluency.
3. We introduce Scientific Concept Description, a challenging new abstractive summarization task, and release a dataset.

The SCD dataset, along with our code, trained models, and human evaluations, will be released at `https://github.com/allenai/pinocchio` shortly after publication.

2 Related work

Pretrained language models have recently taken the top spots on summarization leaderboards (Fabbri et al., 2020; Huang et al., 2020). This includes models like BART (Lewis et al., 2020), PEGASUS (Zhang et al., 2020a), and UniLM (Dong et al., 2019). In a recent large scale evaluation of summarization models, Fabbri et al. (2020) found BART and PEGASUS to be the top performing models. We choose to focus on BART in this work.

It is widely known that SOTA summarization models tend to hallucinate facts (Maynez et al., 2020), and the most closely related works to ours are those on factual summarization. However, we avoid the term "factuality" and instead use "consistency" to denote that the generated summary is supported by the input text. As noted in Maynez et al. (2020), a summary could be hallucinated but still be factually correct. In this work, we aim to improve consistency and reduce hallucinations, which indirectly improves factuality, without directly optimizing for it.

Prior works attempt to improve consistency by correcting already-generated summaries (Dong et al., 2020; Zhu et al., 2021), using a knowledge graph (Zhu et al., 2021), filtering training data (Nan et al., 2021), constraining generation with keywords (Mao et al., 2020), using NLI models (Barrantes et al., 2020; Mishra et al., 2020), among others. Some have focused on the data-to-text setting, which presupposes structured input (Tian et al., 2019; Wang et al., 2020b). Some works control the extractiveness of generations (Song et al., 2020). There have also been multiple works on automatically measuring consistency (Durmus et al., 2020; Kryscinski et al., 2020; Wang et al., 2020a). Matsubara and Singh (2020) noted that hallucinations come from a source-target discrepancy, where many training targets are not fully supported by their source text, and suggested to address it by removing samples with unsupported summaries. We extend their empirical findings with similar measurements on three additional datasets, conjecture that hallucination is unavoidable in such settings, and provide evidence in terms of the lexical statistics of output summaries.

We use beam search for decoding, which has become standard practice for neural seq2seq models (Graves, 2012; Sutskever et al., 2014). Our approach can be viewed as a version of constrained decoding (Hokamp and Liu, 2017) but with dynamically identified constraints and the ability to backtrack. Our constraints come from various model internal signals that indicate attribution to the source text. One such signal is entropy, where Xu et al. (2020) found that low next token entropy indicates the model is copying. Unlike previous work, we do not attempt to imbue models with a new level of textual understanding, but rather show that we can improve consistency of generated text using simple signals based on model internals.

3 Why Do Models Generate Inconsistent Summaries?

In this section, we analyze why models generate inconsistent summaries. Here, we use the definition of *consistent* from Fabbri et al. (2020), i.e., the factual alignment between the summary and the summarized source.

We hypothesize that there are two factors that contribute to inconsistency: 1) the maximum likelihood training and generation strategy used in summarization models, and 2) imperfect training datasets that contain many instances where the gold target is difficult or impossible to deduce from the source. Specifically, we conjecture that in the presence of these two factors, models are guaranteed to hallucinate because they either 1) default to a background distribution of the most common relevant terms during generation or 2) learn spurious correlations between the source and target texts. In either case, the model generates text that is often inconsistent with the inputs.

We present our analysis in terms of a motivating example below, and provide empirical support for it in Sec. 7. The analysis inspires the design of the PINOCCHIO method in Sec. 4.

3.1 Motivating example

Consider the gold summary of an article about a team signing a football player, from XSUM:

`'League Two club Cheltenham Town have signed Hibernian striker Brian Graham on a free transfer.'`

Many of the details in this summary are difficult for a model to predict because they are *not* supported directly by the input passage.[2] For example, the player's first name ("Brian") and position ("striker"), and the lack of signing fee ("free transfer") are nowhere mentioned. This mismatch between typical summary fields and the text available in the input passage is not restricted to summaries about player signings, but is more generally observed across a variety of article types in XSUM and also our new SCD data set.

Achieving a high likelihood on the training dataset requires that the trained models output the aforementioned fields anyway: e.g., in summaries of player signings, from a sample of 43 summaries, 100% mention the player's full name, 88% the player's position, 78% the length of the signing, etc, even though they are often not supported in the source. As a result, the BART summarizer outputs the following summary for the example:

`'League Two side Cheltenham Town have signed Hibernian midfielder Scott Graham on loan until the end of the season.'`

This summary begins nearly identically to the gold, but then outputs the three field values incorrectly (first name, position, and length of the contract).

The errors make sense when you consider the model's calculus for choosing a summary. Consider a single field that can be present or absent in a summary, and make the simplifying assumption[3] that the probability of the most-likely summary with a field value is strictly monotonic in the probability of the field value (see App. B for formal details). In that case, a model that maximizes likelihood will output the field if and only if its best guess of the field value is more probable than the field's absence. In practice, the probability of field absence is often low because training summaries of certain topics reliably cover certain fields, and the best guess probabilities are often higher because the model can do *some* inference to narrow the choice set to a limited and typically peaked distribution (e.g., to a small number of football player positions). Thus, hallucinating a best guess is often preferred by the model—even, in some cases, when the model estimates that the guess is less likely than chance to be correct. In the example, since the estimated probability that the player is a "midfielder" is relatively high ("midfielder" is relatively common, see Fig. 1) and position going unmentioned is rare (about 12% of the time), the model chooses to incorrectly output "midfielder."

Of course, the assumptions in our analysis may not always hold, and hallucination is likely more

[2]The full input passage summarized in this example is in App. C.2.

[3]Note that the PINOCCHIO *method* (Sec. 4) does not depend on this assumption, it is only used here for intuition and ease of analysis.

complex than the single phenomenon analyzed here. But our approach, motivated by the above conjecture, can improve the consistency of summaries in practice. Further, in Section 7 we validate two aspects of our analysis empirically, showing that ground truth training summaries for abstractive summarization do contain unsupported statements, and that summarizers do disproportionately produce more common terms in their output.

4 PINOCCHIO: Constraining Beam Search to Improve Consistency

Inspired by the previous analysis, we introduce PINOCCHIO; a modification to standard beam search for **supported-decoding** (Alg. 1).

Beam search for text generation typically works by adding to a small set of candidate generations one token at a time, keeping the top B generations according to model-predicted likelihood after each prediction timestep. After `<end>` has been predicted in B beams, those B candidates are rescored with a length penalty (Wu et al., 2016), and the best one is chosen as the final output. PINOCCHIO differs from regular beam search only in its use of the set R, which holds a set of disallowed generation paths; if R is always empty, Alg. 1 simplifies to standard beam search. PINOCCHIO modifies the model predicted token scores to avoid inconsistent predictions.

In particular, PINOCCHIO applies a function $f_c(\text{model state}, \text{candidate next generation})$ to the predicted likelihood of the top predicted tokens. If all top predicted tokens for a given timestep are inconsistent according to f_c, PINOCCHIO backtracks by removing the last predicted token from each beam, and predicts again without the ability to predict the removed tokens. The number of times this backtracking occurs η, combined with the average entropy of the token predictions in the final output is a good indicator of whether the model succeeded in producing a good summary or not. Thus, we eliminate generations with multiple backtracks (e.g., $\eta > 2$) and high entropy, as well as individual sentences with high entropy (>2.75) from multi-sentence outputs.

Within this framework, we present an instantiation of f_c based on a set of carefully curated heuristics, determining if a token is allowed to be predicted or not.

Inspired by the aforementioned analysis, f_c consists of a series of binary checks, which take into account both model internals as well as language features. First, it considers the *model confidence* for the current prediction—a high entropy of the token prediction probability distribution over some thresholds may correspond to less certain predictions and thus inconsistent facts. Second, it keeps tracks of the source text with high *attention* scores during the generation process: when the attended texts are semantically or lexically different from the top generated tokens, it suggests a high probability of hallucination. Third, PINOCCHIO also allows tokens that are especially *common*. We develop a total of 8 different binary functions within the three categories above (details in §D.1), and a failure to pass any of the checks may lead to $f_c = 0$ and backtrack during the generation process.

The heuristics do not require additional training steps, and all the associated thresholds or hyperparameters can be determined by manual inspection on a small number of samples (e.g., n=20) from each dataset. Different from prior work Matsubara and Singh (2020), this non-machine learning approach is based on scrutiny of the model generation process. It is easy to execute and more explainable compared to black box models.

Algorithm 1: Supported-decoding

Input: beam size B, generative model M,
  consistency function f_c, vocab V,
  maximumly allowed backtrack count N
priority queue PQ = ["<start>"]*B;
completed generations CG = { };
rejected paths R = { };
backtrack count η = 0;
while |CG| < B **do**
  $C := \{x + v : x \in PQ, v \in V\} - R$;
  $T :=$ top $2B$ items of C scored by M;
  $R := R \cup \{d \in T : f_c(M, d) = 0\}$;
  if $T - R == \emptyset$ **then**
    if $\eta \geq N$ **then**
      `// Stop Generation`
      return { };
    end
    $R := R \cup \{\mathrm{x}[: -1] : x \in T\}$;
    $PQ := \{\mathrm{x}[: -1] : x \in PQ\}$;
    $\eta := \eta + 1$;
    continue;
  end
  $T := T - R$;
  $PQ :=$ top B elements of T according to M not ending in "<end>";
  $CG := CG \cup \{d \in T : d$ scores higher than min in PQ and ends in "<end>"$\}$;
end
return top-ranked element of CG;

5 Tasks and Datasets

We evaluate PINOCCHIO on two distinct summarization tasks: news summarization (XSUM) and scientific concept description (the newly proposed SCD dataset).

5.1 XSUM

XSUM (Narayan et al., 2018) is a popular abstractive news summarization dataset. XSUM is a challenging dataset; the source text frequently does not entail the target text, the target task is not exactly summarization (XSUM is closer to headline generation than summarization), and data is noisy (e.g. there are articles in another language, Welsh). Challenges aside, XSUM is highly regular, as mentioned in Sec. 3. Although this seems to make the task easier, a strong pattern matcher will reproduce dataset patterns (see Tab. 7 for example patterns), *whether or not* it is able to fill in all the details in the pattern correctly.

5.2 Scientific concept description

We introduce the novel task of *scientific concept description* (SCD): automatically generating a brief description of a scientific concept, given the concept name and some papers discussing the concept. SCD training data is inspired by the WikiSum dataset (Liu* et al., 2018), where a concept description (model output) is a Wikipedia intro section and the input is the research papers cited in the Wikipedia page. The dataset is 60K examples with an average input of 319 sentences and average output of 6 sentences per examples. Test data has been manually evaluated to ensure quality. See App. E for the dataset details.

6 Experiments

6.1 Metrics

We rely on human evaluation, as current automatic metrics are unreliable for evaluating factuality (see §6.5). We are not targeting ROUGE metrics (Lin, 2004), but present them for completeness.[4]

For human evaluation, we use standard dimensions of consistency (does the source entail the target?), fluency (is the target grammatical, understandable English?), relevance (does the target contain important information for understanding the source?), and coherence (do the sentences flow together coherently?)[5], with definitions adapted slightly from (Fabbri et al., 2020) via calibration with our annotators. We also decided to rate consistency and fluency on a five-point 1-5 scale, but relevance and coherence on a coarser three-point 1,3,5 scale. See App. E for annotation guidelines.

6.2 Manual evaluation

In Tab. 2, we report manual evaluation results, with each example annotated by one annotator. First, PINOCCHIO improves overall consistency. PINOCCHIO is more consistent 44% and 24% of the time on SCD and XSUM respectively, vs 16% and 13% for BART. Second, on the examples where PINOCCHIO produces no output, BART's output is less factually consistent (0.30 and 0.44 points lower than the full set on SCD and XSUM respectively). Combined, these two wins increase the precision (with respect to complete factual consistency) without hurting recall, yielding an F1 improvement from 0.209 to 0.345 and 0.287 to 0.361 on SCD and XSUM respectively. Third, PINOCCHIO *does* reduce fluency with respect to the base BART model. Fourth, the sentence level entropy filter applied in PINOCCHIO sometimes removes the key first sentence that defines the entity in SCD, resulting in a decrease in relevance.

Pretrained language models are capable of producing incredibly fluent text and prior work on steering them over-optimizes for maximizing the highest likelihood output (Subramani et al., 2019; Subramani and Suresh, 2020). As a result, steering them away from their highest likelihood output as PINOCCHIO does is bound to reduce fluency. Our results suggest that some of this fluency is coming at the cost of factual consistency, as the model has learned how to follow patterns to produce plausible sentences, but not necessarily while sticking to the source text (see §3 and §7.2).

6.3 Automatic evaluation

For completeness, we report ROUGE 1, 2 and L (Tab. 3). We note here that there is a substantial difference in the effect of PINOCCHIO on ROUGE between the two datasets, with a large drop on XSUM (see §7.2 for discussion).

6.4 Inter-annotator agreement

In Tab. 4, we report various inter-annotator agreement measures. We had three expert annotators, and the agreement stats are averaged between all pairs of annotators, on a set of 30 examples (15 from each model) from each dataset. For model

[4] https://github.com/Yale-LILY/SummEval

[5] Coherence not used on XSUM as targets are 1 sentence

Method	Dataset	% Cons.=5	% Cons.= 4/5	Cons.	Flue.	Rele.	Cohe.
BART (n=282)	XSUM	0.287	0.709	3.908	**4.794**	4.887	-
PINOCCHIO (n=211)	XSUM	0.422	0.82	**4.19**	4.649	4.886	-
BART (n=268)	SCD	0.209	0.552	3.612	**4.537**	**4.925**	4.619
PINOCCHIO (n=207)	SCD	0.396	0.768	**4.082**	4.338	4.816	4.585

Table 2: Human evaluation of models. PINOCCHIO improves consistency significantly, while decreasing fluency slightly. For the 4 evaluation metrics, significant (Mann–Whitney U test, p<0.01) differences are bolded. Cons.=Consistency, Flue.=Fluency, Rele.=Relevance, Cohe.=Coherence

Method	Dataset	# Samples	R1	R2	RL
BART	XSUM	11333	0.444	0.210	0.354
BART*	XSUM	8345[1]	0.442	0.207	0.349
PINOCCHIO	XSUM	8345	0.431	0.196	0.338
BART	SCD	4647	0.389	0.174	0.277
BART*	SCD	2335	0.402	0.190	0.292
PINOCCHIO	SCD	2335	0.394	0.182	0.286
BART	CNN/DM	10990	0.438	0.209	0.372
BART*	CNN/DM	10943	0.438	0.209	0.372
PINOCCHIO	CNN/DM	10943	0.438	0.209	0.372

[1] Because PINOCCHIO can elect to skip in certain cases, we report two scores for BART model outputs: for all test samples, and for the samples where PINOCCHIO generates results.

Table 3: Rouge scores on different datasets with and without using PINOCCHIO. Datasets with higher abstractiveness (e.g., XSUM and SCD) may suffer from higher ROUGE drops when PINOCCHIO is used.

Metric	Dataset	Cons.	Flue.	Rele.	Cohe.
tau	XSUM	0.60	0.84	-	-
exact	XSUM	0.66	0.89	0.96	-
compare	XSUM	0.69	0.82	1.0	-
compare~	XSUM	1.0	1.0	1.0	-
tau	SCD	0.55	0.43	0.20	0.53
exact	SCD	0.55	0.69	0.93	0.80
compare	SCD	0.67	0.64	0.86	0.64
compare~	SCD	0.95	0.98	1.0	0.98

Table 4: Mean agreement metrics between all pairs of annotators. tau=Kendall's tau, exact=exact agreement, see §6.4 for compare and compare~. The very low/null correlation values are due to low variance in relevance.

comparison, the most important metrics are the "compare" metrics, which measure how often the annotators agree on which model's output is better for a given example. The "compare" metric is the fraction of examples for which the pair of annotators agree on which model's output is better or both say the outputs are equivalent. The "compare~" metric is similar but more lenient, as it only counts as disagreement the examples where one annotator says one model is better, and the other annotator says the opposite. These kinds of strong disagreements are very rare in our data, suggesting that the relative comparisons between models in our experiments are reliable.

6.5 Comparison against existing correctors and factuality metrics

We also compare with three recent methods for automatically correcting summaries or measuring their factuality. Here we evaluate on XSUM, which we expect to be more suitable for these methods (each were evaluated on XSUM in previous work, whereas SCD is out of domain). First, we compare against Zhu et al. (2021), a recent seq2seq fact corrector (FC) that incorporates OpenIE (Angeli et al., 2015) and knowledge graph embedding. We take the output of their strongest model (UniLM (Dong et al., 2019)+FC) on the XSUM test set and find that it changes only ~5% of examples, and that the net improvement rate of the changes is 15% (see App. F for details). This corresponds to an improvement on <1% of the full XSUM test set. By contrast, our experiments in the previous section show that PINOCCHIO yields an improvement on ~8.5% of XSUM, more than a factor of eight higher.

Finally, we assess two representative automatic factuality metrics, FactCC (Kryscinski et al., 2020) and FEQA (Durmus et al., 2020). FactCC trains a <source, summary sentence> classifier; FEQA generates/answers questions from the summary, checking if answers are the same when using the source. We find neither metric suitable for our highly abstractive setting; each has low agreement with our XSUM annotations (Tab. 5), a result in line with a very recent evaluation of factuality measures (Pagnoni et al., 2021).

Metric	FactCC	FEQA
tau	-0.02	0.233
compare$\neq$	0.528	0.585
mean/σ pairwise ties	1.354/1.464	0.108/0.096
mean/σ pairwise not ties	1.699/1.518	0.113/0.1

Table 5: Agreement between automated metrics and our annotations. tau represents Kendall's tau, compare$\neq$ denotes agreement with the annotator on which model is better, *when the annotator did not rate the models as equivalent*, "Mean/σ pairwise ties" gives the mean/std of absolute value of difference between the metric's rating for each model, for pairs where the annotator rated the models as the same, and "Mean/σ pairwise not ties" is the same but for pairs where the annotator rated the models as different. A well-calibrated metric should have mean near zero and low standard deviation when the models are annotated as equivalent. We find the automated metrics exhibit low agreement with our annotators.

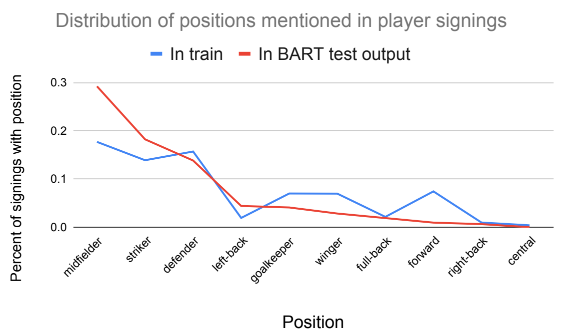

Figure 1: Distribution of positions in summaries about player signings in train vs. BART output. BART output is more peaked at positions more common in train, suggesting BART defaults to these when no position is supported by the source.

7 Discussion

7.1 Empirical validation of the intuition motivating PINOCCHIO

We now present two empirical analyses to verify the intuition sketched in Section 3. First, we verify our claim that the ground truth summaries in our data sets contain unsupported terms (Table 6). We define *Dataset Abstractiveness* as the ratio of n-grams that appear in the summary but not in the source text. The two abstractive datasets (XSUM and SCD) show high abstractiveness, with approximately half or more of the terms in the summaries not appearing in the source. Of course, a lack of lexical overlap could arise from summaries stating supported information but in different terms from the source. Thus, we also manually examine twenty examples for XSUM and CNNDM and ten for SCD and measure the fraction that are not directly supported by the source.[6] This fraction is substantial (18-24%) for the abstractive datasets, but much smaller (2%) for the more extractive CNN/DM dataset. Finally, η, the number of times our proposed method BART + PINOCCHIO (discussed in the next section) *backtracks*, which is a measure of how often the method estimates that generated tokens are unsupported, also correlates with the abstractiveness measures.

We also verify one expected consequence of our hypothesized mechanism of hallucination. If indeed BART is defaulting to a background distribution of field values (based on frequency in the training summaries), then we would expect the more frequent training values to become even more probable in BART's output, as the model defaults to these as best guesses. We do observe this effect for positions in player signings, as shown in Fig. 1. It is notable that while this distribution is more peaked, it is not entirely concentrated on the most-likely field value, suggesting that the model has learned spurious correlations that lead it to output other more rare field values, even when unsupported.

More generally, we also observe a similar bias across all n-grams; compared to the original ground truth summaries, the BART output tends to be less heavy-tailed, including disproportionately more of the high-likelihood n-grams. We show this by plotting the n-gram frequency distributions (which follow a power law) on a log-log scale in Fig. 2. The BART output generally has a less negative slope than the ground truth distribution on these plots. Our BART + PINOCCHIO method results in a distribution that is closer to the ground truth, for 2- and 3-grams.

7.2 Patterns and hallucination

We now discuss some qualitative aspects of our results. First, we need to discuss the substantial drop in ROUGE on XSUM. As alluded to in §3, we believe this is due to a pervasive regularity in the XSUM dataset, which BART is able to capture very well. In Tab. 7, we show the top examples sorted by ROUGE-L difference between BART and PINOCCHIO, along with a hand-crafted regex

[6] This annotation task can be challenging and subjective especially for the SCD dataset, see appendix §C for details.

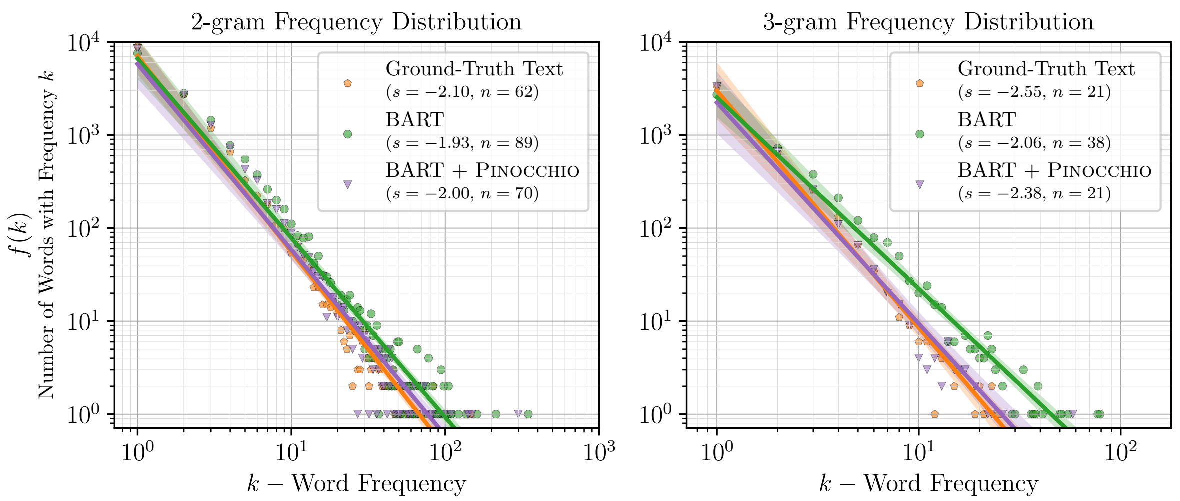

Figure 2: Comparing the n-gram frequency distribution on the XSUM Dataset for generated, versus ground truth sources. The default BART model outputs (in green) over-represent frequent n-grams (bottom right of the distribution), but PINOCCHIO is closer to the ground-truth. Results in the SCD dataset are similar. The slope of the linear fits for ground-truth text and BART generations are significantly different ($p \ll 0.05$, ANCOVA) while those between ground-truth and BART + PINOCCHIO generations are not ($p > 0.05$) for both 2-gram and 3-grams.

	Dataset Abstractiveness					Human Annotated Unsupported Words		BART+PINOCCHIO
Dataset	1-gram	2-gram	3-gram	4-gram	Avg.	% Unsupported Words	IAA - Cohen κ	Avg. η per Successful Generation
CNN/DM[1]	17.18%	58.44%	78.06%	86.71%	60.10%	1.57%	0.571	0.0003
XSUM	49.88%	89.65%	98.13%	99.60%	84.31%	17.78%	0.728	0.1541
SCD	60.16%	88.97%	96.81%	98.61%	86.14%	23.84%	0.414	0.2300

[1] We report the scores for the CNN / Daily Mail dataset (See et al., 2017; Hermann et al., 2015) for comparison because it is highly extractive.

Table 6: Analysis of the abstractiveness of three summarization datasets. The abstractive XSUM and SCD data sets contain a substantial fraction of unsupported words, measured in terms of either automated n-gram overlap measures or manual annotation. Also, BART+PINOCCHIO performs more backtracks η on datasets with high abstractiveness scores.

matching the example, how many times it matches gold examples from the training and validation set, how many times it matches BART predictions on the test set, and how many of those predictions are completely factually consistent. Most of these examples straightforwardly map to patterns of text that occur in the training data. We also see that test set predictions matching these patterns are largely not consistent. As discussed in §3, this is because BART assigns high likelihood to the general pattern, but guesses to fill in the details. Some of these patterns are straightforward to identify, but many are likely to be more complicated. Broadly speaking, XSUM contains a lot of regularity in the mapping between the source topic, phrases, and vocabulary used in the target summary. BART exploits exactly this. PINOCCHIO steers the model away from the patterns, which are often not supported by the source text, which lowers ROUGE.

A related question is if BART trained on XSUM applies facts learned during training correctly. Does it learn that Antonio Conte is the coach of the Italian football team, thus someone named "Conte" who coaches the Italian team is Antonio Conte? Or does it merely learn the first name most commonly associated with "Conte" in train is "Antonio", and so everyone named "Conte" is Antonio Conte? [7] It is difficult to assess this automatically, so we present an example of BART's tendency to guess world knowledge. We create one three-sentence source, "Sometime last week, a fire burned down a <BUILDING>, killing a number of people. The fire took place in <LOCATION>. Investigators believe at least four people to be missing.", filling in the blanks with three made up locations and three building types. BART produces plausible but inconsistent summaries. Nine out of nine outputs

[7] Experiments with this example strongly suggest the latter.

BART generation	Manual pattern	Train/val	Predicted	Consistent
A 70-year-old man who died after being hit by a car in Monmouthshire has been named by police.	.*year-old.*who died.*named.*	47	4	0
Chinese businessman Dr Tony Xia has completed his £52m takeover of Championship club Aston Villa.	.*Tony Xia.*Aston Villa.*| .*Aston Villa.*Tony Xia.*	7	1	0
All pictures are copyrighted.	.*All pictures are copyrighted.*	44	4	4
Forfar Athletic extended their lead at the top of Scottish League Two to five points with a 3-0 win over Berwick Rangers.	.*extended.*top.*points.*win.* .*Forfar Athletic.*top of Scottish League Two.*	9 10	3 1	0 0

Table 7: Top-5 BART generations, by ROUGE-L gain over PINOCCHIO (#2 is excluded; it doesn't match an obvious pattern and is factually consistent). In all examples, BART clearly memorized training patterns and guesses the details in at least 3 (the 3rd output is memorized from noise in XSUM), which is not strongly penalized by ROUGE.

hallucinate the location, eight discuss arrests or hospitalizations, and three mention the police or fire service reporting the details of the situation. These characteristics are all due to biases present in the training data. Locations are often abstracted, reported fires often result in someone being arrested or hospitalized, and they are usually reported by authorities. We present this example as evidence that BART is not learning how to reliably apply commonsense and learned facts, but rather, is naively reproducing patterns and word associations.

7.3 Error analysis

To provide insight into dominant error types, we sample 20 inconsistent PINOCCHIO generations from SCD evaluation, identifying three common error causes, each occurring in ~20% of the samples: 1) Incorrect paraphrasing or omission of meaning-changing information (e.g. X has a long history of being used for Y vs. X is the model of choice for Y) 2) Incorrect treatment of entities as coreferent/synonymous 3) Difficulty with heavy mathematical notation. Targeting each of these in a generative model is a promising future direction.

8 Conclusion

In this work, we present PINOCCHIO, a simple, no-additional-machine-learning required, method for reducing hallucination in generative encoder-decoder models. PINOCCHIO provides a substantial lift in consistency, with only a small decrease in fluency. We analyze why existing summarizers hallucinate, showing that distantly supervised abstractive summarization datasets can contain unsupported target summaries, and presenting evidence for our conjecture that models that maximize likelihood trained on such data will tend to hallucinate. We also show that existing factuality metrics are insufficient, and further explore how patterns in the training dataset can produce misleading results on the test set. We also introduce the task of scientific concept description and release a Wikipedia-based dataset for this task.

We would like to clearly acknowledge the limitations of our approach. PINOCCHIO does not add new learned behavior to the model, using simple heuristics and single-step backtracking to steer the model towards more consistent output. The heuristics have settings that require some adaptation for each data set. Further, preliminary experiments suggest that the settings that were effective for BART do not simply work out of the box for another summarizer, PEGASUS. We hope the approach and insights in this paper help spur further development of models that generate consistent text, and datasets where the source entails the target and spurious patterns are minimal.

Acknowledgement The authors would like to thank Noah Smith and Kyle Lo for helpful comments. This material is based upon work supported by NSF Grant OIA-2033558, NSF RAPID grant 2040196, ONR grant N00014-18-1-2193, and the Allen Institute for Artificial Intelligence (AI2).

References

Gabor Angeli, M. Johnson, and Christopher D. Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *ACL*.

Mario Barrantes, Benedikt Herudek, and R. Wang.

2020. Adversarial nli for factual correctness in text summarisation models. *ArXiv*, abs/2005.11739.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. 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 3615–3620, Hong Kong, China. Association for Computational Linguistics.

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

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Yue Dong, Shuohang Wang, Zhe Gan, Yu Cheng, Jackie Chi Kit Cheung, and Jingjing Liu. 2020. Multi-fact correction in abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9320–9331, Online. 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 *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5055–5070, Online. Association for Computational Linguistics.

Alexander R Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2020. Summeval: Re-evaluating summarization evaluation. *arXiv preprint arXiv:2007.12626*.

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

Alex Graves. 2012. Sequence transduction with recurrent neural networks. *ArXiv*, abs/1211.3711.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *NIPS*, pages 1693–1701.

Chris Hokamp and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1535–1546, Vancouver, Canada. Association for Computational Linguistics.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Dandan Huang, Leyang Cui, Sen Yang, Guangsheng Bao, Kun Wang, Jun Xie, and Yue Zhang. 2020. What have we achieved on text summarization? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 446–469, Online. Association for Computational Linguistics.

Daniel King, Doug Downey, and Daniel S. Weld. 2020. High-precision extraction of emerging concepts from scientific literature. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*.

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, Online. Association for Computational Linguistics.

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

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *ACL 2004*.

Peter J. Liu*, Mohammad Saleh*, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. 2018. Generating wikipedia by summarizing long sequences. In *International Conference on Learning Representations*.

Yang Liu and Mirella Lapata. 2019. Hierarchical transformers for multi-document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5070–5081, Florence, Italy. Association for Computational Linguistics.

I. Loshchilov and F. Hutter. 2017. Fixing weight decay regularization in adam. *ArXiv*, abs/1711.05101.

Yuning Mao, X. Ren, Huai zhong Ji, and Jiawei Han. 2020. Constrained abstractive summarization: Preserving factual consistency with constrained generation. *ArXiv*, abs/2010.12723.

Yoshitomo Matsubara and Sameer Singh. 2020. Citations beyond self citations: Identifying authors, affiliations, and nationalities in scientific papers. In *Proceedings of the 8th International Workshop on Mining Scientific Publications*, pages 9–20, Wuhan, China. Association for Computational Linguistics.

Kazuki Matsumaru, Sho Takase, and Naoaki Okazaki. 2020. Improving truthfulness of headline generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1335–1346, Online. Association for Computational Linguistics.

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.

Anshuman Mishra, Dhruvesh Patel, Aparna Vijayakumar, Xiang Li, Pavan Kapanipathi, and Kartik Talamadupula. 2020. Looking beyond sentence-level natural language inference for downstream tasks. *ArXiv*, abs/2009.09099.

Feng Nan, Ramesh Nallapati, Zhiguo Wang, Cicero Nogueira dos Santos, Henghui Zhu, Dejiao Zhang, Kathleen McKeown, and Bing Xiang. 2021. Entity-level factual consistency of abstractive text summarization. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2727–2733, 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.

Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. 2021. Understanding factuality in abstractive summarization with frank: A benchmark for factuality metrics. *ArXiv*, abs/2104.13346.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

Kaiqiang Song, Bingqing Wang, Zhe Feng, Ren Liu, and Fei Liu. 2020. Controlling the amount of verbatim copying in abstractive summarization. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8902–8909. AAAI Press.

Nishant Subramani, Samuel R. Bowman, and Kyunghyun Cho. 2019. Can unconditional language models recover arbitrary sentences? In *NeurIPS*.

Nishant Subramani and Nivedita Suresh. 2020. Discovering useful sentence representations from large pretrained language models. *ArXiv*, abs/2008.09049.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*.

Ran Tian, Shashi Narayan, Thibault Sellam, and Ankur P. Parikh. 2019. Sticking to the facts: Confident decoding for faithful data-to-text generation. *ArXiv*, abs/1910.08684.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020a. 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, Online. Association for Computational Linguistics.

Zhenyi Wang, Xiaoyang Wang, Bang An, Dong Yu, and Changyou Chen. 2020b. Towards faithful neural table-to-text generation with content-matching constraints. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1072–1086, Online. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Jiacheng Xu, Shrey Desai, and Greg Durrett. 2020. Understanding neural abstractive summarization models via uncertainty. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6275–6281, Online. Association for Computational Linguistics.

Jingqing Zhang, Y. Zhao, Mohammad Saleh, and Peter J. Liu. 2020a. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *ICML*.

Yuhao Zhang, Derek Merck, Emily Tsai, Christopher D. Manning, and Curtis Langlotz. 2020b. Optimizing the factual correctness of a summary: A study of summarizing radiology reports. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5108–5120, Online. Association for Computational Linguistics.

Chenguang Zhu, William Hinthorn, Ruochen Xu, Qingkai Zeng, Michael Zeng, Xuedong Huang, and Meng Jiang. 2021. Enhancing factual consistency of abstractive summarization. In *NAACL*.

A Example full text

A.1 Example from Table 1

Police said the 14-year-old reported feeling unwell and required hospital treatment. He was later discharged from hospital and is recovering at home. The incident happened in Holywood, County Down, on Saturday. The PSNI said the tablets were "as yet unidentified" but warned of the "potential dangers" they posed. The 17-year-old, has been charged with possessing a Class A controlled drug with intent to supply; possessing a Class B controlled drug with intent to supply; possession of a Class A controlled drug; possession of a Class B controlled drug and supplying a Class A controlled drug. He is due to appear at Newtownards Youth Court on 14 February.

A.2 Player signings

The 29-year-old Scot has signed a two-year contract with the Gloucestershire outfit. Prior to joining Hibs in August 2016, Graham had spells at six other Scottish sides, including Dundee United, St Johnstone and Ross County. He will be available for Saturday's league visit of Crawley Town, subject to receiving international clearance. Find all the latest football transfers on our dedicated page.

B Mathematical Details of Hallucination Analysis

Formally, a summarization model is defined by a distribution $P(S|P)$ over output textual summaries S conditioned on an input passage P. We assume that the summarization system aims to maximize the probability of the summary S given the text passage, i.e. it outputs $\arg\max_S P(S|P)$. While in practice (including in our experiments), summarization models use imperfect search procedures like beam search to find high-likelihood generations, and may rescore complete generations using factors other than likelihood (like length), in this analysis we ignore these details and assume the generator simply maximizes likelihood. Analyzing the impact of more complex generation aspects is an item of future work.

Let $F(S)$ be a function denoting the value of a given "field" in the summary S, equal either to some string value or to $\emptyset$ if the field does not occur in S. A "field" is a typical piece of information that is often mentioned in a summary of a given topic (e.g., participating teams, in a summary of a sporting event; or the university where an idea was developed, in a scientific concept description). Then the model's distribution over a field value for a given passage is $P(F = \mathbf{f}|P) = \sum_S P(F(S) = \mathbf{f}|P)$.

Our analysis uses the following assumption:

Assumption **A1**: The model's most likely summary probability is strictly monotonic in the probability of its included field values. That is, whenever:

$$P(F = \mathbf{f}|P) > P(F = \mathbf{f}'|P) \tag{1}$$

then

$$\max_S P(S, F(S) = \mathbf{f}|P) > \max_S P(S, F(S) = \mathbf{f}'|P) \tag{2}$$

That is, when the model thinks a field value is more likely in a summary for a given passage, then it can find a more likely summary that uses that field value. This assumption seems likely to hold often in practice (for example, we would expect that by simply swapping out a less likely field value in a summary for a more likely one, we would often arrive at a more probable summary).

The observation used in the analysis in Section 3 is then:

Given a passage p, a field F, and a summarization model $P(S|P)$, if assumption A1 holds, then a generator that maximizes likelihood will choose to output $\mathbf{f} = \arg\max_{\mathbf{f}} P(F = \mathbf{f}|P)$ for the field's value (or omit the field, if $\mathbf{f} = \emptyset$).

Proof: Consider an alternative field value $\mathbf{f}'$, which may be $\emptyset$, with $P(\mathbf{f}|P) > P(\mathbf{f}'|P)$. Let $S^* = \arg\max_S P(S, F(S) = \mathbf{f}|P)$ and $S' = \arg\max_S P(S, F(S) = \mathbf{f}'|P)$. Since $P(\mathbf{f}|P) > P(\mathbf{f}'|P)$, by A1 we know that $P(S^*|P) > P(S'|P)$. Since the generator maximizes likelihood, it will output S^* (which includes the field value $\mathbf{f}$) instead of S' or any other summary including $\mathbf{f}'$.

C Manual Examination of the Unsupported Dataset Samples

Identifying parts of a summary that are not supported by the source document is a challenging annotation task. In this section, we explain how we formalize this task as a binary token tagging problem, and we show one example that illustrate the difficulty of annotation.

C.1 Annotating unsupported words

Naturally, words that appear only in the summary but not the source document tend to have a higher chance of being "hallucinated", and vice versa. Hence, we select such words from the summary, and the goal is try to identify whether the meaning of these words can be deduced from the source documents. Compared to the automated measurements, the manually inspected labels are considered to be a better approximation of the true abstractiveness of the dataset or the samples.

C.2 One challenging example

In practice, understanding the source document involves multiple (common sense) reasoning steps and subjective judgements.

Considering the following document text:

`'ABC of allergies: Venom allergy Stings from bees and wasps, the most common stinging insects in Britain, can cause severe allergic reactions, including anaphylaxis. Coroners' data suggest that an average of four deaths from bee or wasp stings occur each year in the United Kingdom, but this is almost certainly an underestimate because venom anaphylaxis is not always recognised as the cause of death'`

For one sentence in the summary, we highlight the words that do not appear in the source in red:

`'The stings of most of these species (Bees) can be quite painful, and are therefore keenly avoided by many people.'`

The source text mentions several dangerous aspects of bee stings, but whether it can be concluded that they are avoided by many people (a plausible commonsense implication) is subjective to judge, and annotators often had differing opinions on these judgments.

D PINOCCHIO Details

D.1 Heuristics in f_c

We develop 8 binary checks that constitute the heuristics for f_c, which fall into three categories. Two categories use model internals, *model confidence* and *source text attribution* for the predicted token. The third category uses language features, allowing generations that are *common words*.

Model confidence

- entropy of next-token distribution < τ for a token in the top 2 predictions
- from the top 10 predicted next tokens, the number that match a top 5 attended-to piece of source text[8] is $>= \frac{1}{2}(10-$the number that are stopwords$)$

Source text attribution

- the most attended-to piece of the source text contains the predicted token
- 3 out of the top 5 attended-to pieces of the source text contain the predicted token
- sum of the attention scores of the attended-to pieces of source text (out of the top 5) that contain the predicted token is greater than $\frac{1}{3}$ of the sum of the top 5 attention scores
- max cosine similarity between the embedding of the predicted token and that of any word in the top 5 attended-to pieces of source text is greater than 0.15 (and the word is not capitalized or a number word)*[9]

Common word

- predicted token is a stopword*
- prediction matches[10] one of the top 5 predictions of roberta-base[11]

All of the components and hyperparameters above were determined via inspection on a small number of samples (e.g., n=20) from the XSUM and SCD dataset. In the subsequent sections we detail the configurations of the parameters on each dataset.

D.2 XSUM modeling details

For configuration of PINOCCHIO for XSUM, we set $\tau = 1.0$ and do not use the optional stopword condition, in order to accommodate the highly abstractive nature of the XSUM dataset and attempt to prevent the use of stopwords in hallucinations.

One other important detail is that XSUM has a surprising property with respect to first names. If a person appears in the source as "Mr/Ms" X, and also in the headline, they *always* appear as <FIRST NAME> X in the headline. This leads to BART *always* guessing the first name of a person, frequently incorrectly. Our f_c often identifies the first name as unsupported, but because BART is essentially

[8] All reference to "top attended-to pieces of the source text" means a max across locations in the source text across attention heads in the final layer of the decoder's cross-attention, and a 10-wordpiece window around the attended-to location.

[9] Items marked with an asterisk * are optional.

[10] For all string matching, we lemmatize first.

[11] https://huggingface.co/roberta-base

unable to predict anything other than a first name in this situation, it is unable to recover from this error. For this reason, when an unsupported token is identified as a name using spaCy (Honnibal et al., 2020), we deterministically replace it with Mr/Ms.[12]

D.3 SCD modeling details

For SCD, the source consists of full papers and is too long to input to BART directly, so we train a separate BERT-based model to extractively rank chunks of the input text based on predicted ROUGE-L F1 score against the target text. This setup of ranking extractive chunks and then passing them to an abstractive model is similar to prior work on long text summarization (Liu and Lapata, 2019). We pass the concept name/aliases and each chunk of text to rank to SciBERT-base (Beltagy et al., 2019), with a final linear layer to predict the ROUGE-L score. We then finetune BART, with the ranked extractive chunks as source, again concatenated with the concept name/aliases. For inference, we also filter the chunks to those that include the concept name or an alias.

Beam search parameters We use standard parameters for the beam search of min_length=5, max_length=500, no_repeat_ngram_size=3, length_penalty=2.0, and num_beams=6.

Extractive ranker for descriptions The extractive ranker uses SciBERT[13], followed by a linear layer, and is trained with MSE loss. We also use dropout of 0.1. We train on chunks containing three sentences, and use the average ROUGE-L as the label. To reduce the size of the training set, for each target description, we select the top 5 and bottom 5 chunks by ROUGE-L, and an additional 5 random chunks from the middle. We train for 3 epochs, with a batch size of 1, 8 gradient accumulation steps, and the AdamW (Loshchilov and Hutter, 2017) optimizer, with weight decay 0.01, and a slanted triangular learning rate scheduler with peak learning rate 5e-5.

D.4 Finetuning BART on descriptions

BART was finetuned with the standard settings,[14] a batch size of 4 with 8 gradient accumulation steps, for 10 epochs, selecting the epoch 5 model based on validation loss. The same optimizer as above was used, with 500 warmup steps. The model was trained for 5.5 hours on 3 NVIDIA Quadro RTX 8000s. We additionally filter out examples that have a target length less than 150 characters, and examples where the source and target have less than 0.2 token overlap.

For configuration of PINOCCHIO for SCD, we set τ=0.75 and do not use the optional cosine similarity condition, to encourage more extractiveness.

E SCD Dataset Construction Details

E.1 Wikipedia intro sections

We take the first section of a Wikipedia article to be its "intro" section, and also include sections with definitional headers (Introduction, Definition, Uses, Description, Function, Overview) in our dataset.

E.2 SCD training corpus

Training an SCD system requires a large set of ground-truth descriptions. Inspired by the WikiSum dataset (Liu* et al., 2018), we construct our training set using Wikipedia intro sections (see App. C for details) as the target descriptions,[15] with the papers cited in each description as source text. To remove intractable examples, we filter out those with lower than 0.15 ROUGE-1 recall between the cited papers and the target Wikipedia description. The dataset is split into train/dev/test with 47570/5989/5839 examples. Examples have 2.4 source documents with a total of 319 sentences on average and target descriptions averaging 6 sentences each. We are able to extract body text for ~57% of the cited papers, and use just the titles and abstracts of the remainder.[16]

E.3 SCD test corpus

The motivating use case for the SCD task is automatically generating a high-quality encyclopedia for the long tail of scientific knowledge presented in papers. As a result, we construct our SCD evaluation examples not from Wikipedia, but instead from a much broader set of scientific concepts mined from computer science papers using

[12] For real applications, we suggest using a gender neutral honorific, as gender is not possible to infer using first names

[13] https://huggingface.co/allenai/scibert_scivocab_uncased

[14] https://huggingface.co/facebook/bart-large/blob/main/config.json

[15] English Wikipedia 4/1/20 dump processed with https://github.com/spencermountain/dumpster-dive

[16] Due to copyright restrictions we only release the open access subset of this corpus which is smaller (~24% of the papers with body text), but in many cases the PDF may be available (just not licensed for redistribution), and we provide metadata to assist with this (e.g., Unpaywall or arXiv links).

ForeCite (King et al., 2020). This set lacks gold target descriptions, so it requires manual evaluation.

Training on surrogate data that differs somewhat from the intended use case but can be obtained at scale is common in summarization research (e.g. abstracts as paper summaries (Cohan et al., 2018); headlines as news summaries (Narayan et al., 2018)). In our case there are two major discrepancies between train and test: the textual domain (train is mostly biomedical, test is largely computer science), and the level of supporting text (the Wikipedia-cited training inputs often have less support for the concept description than the ForeCite-mined test inputs do, as ForeCite pairs concepts with their likely introducing paper(s)).

F Annotation instructions

- Consistency
  - 1: completely made up
  - 2: some phrases supported, but largely made up
  - 3: some full details correct, but key details made up
  - 4: minor details not fully supported (e.g. acronym wrong, location abstracted a bit wrong)
  - 5: fully supported
  - Other notes: An unresolved "it" should be assumed to refer to the main concept. If this makes it not factual, that counts against consistency, otherwise it counts against coherence.
- Coherence
  - 1: all sentences/phrases don't make sense together
  - 3: some sentences/phrases don't make sentence together, separate from whether they are factual
  - 5: no issues with how phrases/sentences are put together
- Fluency (at the sentence level)
  - 1: not fluent English to the point that it is impossible to understand/meaningless
  - 2: not fluent English to the point that it is very hard to understand
  - 3: semi fluent English (including major fluency errors resulting from copying source text), but still largely understandable
  - 4: Mostly fluent English (including minor fluency errors resulting from source text), does not impact understanding
  - 5: Fluent English
- Relevance
  - 1: off-topic
  - 3: mostly on-topic or seems to be missing an actual statement of what the concept is (or for news, what the article is about)
  - 5: on-topic and contains the key statement of what the concept is (or for news, what the article is about)

G UniLM+FC comparison details

Model output downloaded from `https://drive.google.com/file/d/1blmmJvniToN1yedoWUH3u0SNtXnMVDAs/view?usp=sharing` on 03/23/21. We consider an output "changed" by FC if it is not a prefix match for the original UniLM output, after lowercasing and removing spaces and apostrophes. Many FC-corrected examples seem to simply cutoff the end of the generated text. We choose to not count these as "changed." There are 579 such cases. Given this criteria, FC changes 594 examples in the XSUM test set, and we sample 100 of these for evaluation. FC makes very minimal edits, so it is straightforward to identify whether the edit is an improvement or not. The net improvement is the number of increases in consistency minus the number of decreases in consistency.