Training Dynamics for Text Summarization Models

Anonymous ACL submission

Abstract

Pre-trained language models (e.g. BART) have shown impressive results when fine-tuned on large summarization datasets. However, little is understood about this fine-tuning process, including what knowledge is retained from pre-training models or how content selection and generation strategies are learnt across iterations. In this work, we analyze the training dynamics for generation models, focusing on summarization. Across different datasets (CNNDM, XSUM, MEDIASUM) and summary properties, such as abstractiveness and hallucination, we study what the model learns at different stages of its fine-tuning process. We find that a propensity to copy the input is learned early in the training process consistently across our domains of interest. On the other hand, factual errors, such as hallucination of unsupported facts, are learnt in the later stages, though this behavior is more varied across domains. Based on these observations, we explore complementary approaches for modifying training: first, disregarding high-loss tokens that are challenging to learn and second, disregarding low-loss tokens that are learnt very quickly. We show that these simple training modifications allow us to configure our model to achieve different goals, such as improving factuality or improving abstractiveness.[1]

1 Introduction

Transformer-based pre-training (Lewis et al., 2020; Zhang et al., 2020) has led to substantial improvements in the performance of abstractive summarization models. This pre-training and fine-tuning paradigm has been widely studied with respect to what training datasets, model sizes and other hyperparameters are needed to optimize task-specific evaluation metrics, such as perplexity or ROUGE for text generation. However, abstractive summarization is a complex task involving several components such as content selection and rewriting that are performed implicitly by end-to-end models such as BART (Lewis et al., 2020). Currently, we have little insight into this aspect of the fine-tuning process, namely what "skill" or behavior is learnt at which stage of the training process.

Recent work (Schuster et al., 2019; Utama et al., 2020a) has studied training dynamics for sequence classification tasks such as NLI and fact verification, demonstrating how these can be leveraged to mitigate dataset biases. However, similar analyses for text generation tasks have not been explored. In general, generation models are expected to differ from these classification frameworks due to the difference in task formulations and mismatch between training and inference (teacher forcing).

In this paper, we make the first attempt at understanding the fine-tuning process of large pre-trained language models for summarization. We study two essential components of abstractive summarization models, abstractiveness and factual consistency, and investigate when each of these is learned during fine-tuning. Experiments are conducted on three different summarization datasets: XSUM (Narayan et al., 2018), CNNDM (Hermann et al., 2015; Nallapati et al., 2016) and MEDIASUM (Zhu et al., 2021) to study these properties across a range of datasets.

Our findings are threefold: First, we find that easy-to-learn skills such as copy behavior are acquired very early in the fine-tuning process. In fact, for datasets that have a high fraction of extractive summaries, the summarization models tend to overfit to these easier examples, effectively ignoring harder examples in the dataset. Next, we investigate how factual correctness of summaries evolves with the fine-tuning process, juxtaposing it against other factors such as abstractiveness and dataset quality. In particular, we find that while non-factuality and abstractiveness are roughly proportional to each other, longer training on noisy datasets can significantly hurt factuality.

[1] All code and model checkpoints will be released.

Finally, we show that insights from these training dynamics can be leveraged to optimize along target summarization goals like factuality or abstractiveness. We extend prior work on loss truncation (Kang and Hashimoto, 2020), using token sub-sampling to dynamically modify the loss computation during training to alter the learnt behavior of summarization models. In particular, we show that we can substantially improve the factuality of summarization models trained on noisy datasets (e.g. XSUM) by downweighting high-loss tokens while preserving the high level of abstractiveness. Conversely, downweighting low-loss tokens under the same framework allows us to significantly improve the abstractiveness of generated summaries compared to the baseline models for relatively extractive datasets (e.g. CNNDM and MEDIASUM).

2 Learning Dynamics

2.1 Datasets and Setup

We study learning dynamics for summarization models trained on three English-language news datasets: (1) **XSUM**: an "extreme" summarization dataset with single-sentence and highly abstractive summaries (2) **CNN/DAILYMAIL**, a multi-sentence summary dataset with a considerably lower degree of abstraction. (3) **MEDIASUM**, a media interview summarization dataset with a degree of abstraction closer to CNNDM than XSUM. We focus on the NPR-specific subset of this dataset which contains multi-sentence summaries. These datasets were selected because of the diversity of their respective reference summaries along properties such as lexical overlap, length, and lead-bias within the news summarization domain.

Experiments are performed using BART-LARGE and PEGASUS-LARGE as the base models. For each dataset, the model checkpoints are saved periodically (every 2k steps for XSUM and MEDIASUM, every 1k steps for CNNDM) and analyzed at 10 different stages of the fine-tuning process (9 intermediate checkpoints + final model). Training details are in Appendix A. We probe the model behavior at each checkpoint via two types of signals:

1. **Model-generated summaries**: For each dataset, we randomly sample 800 (article, reference summary) pairs from the development set. At each checkpoint, we generate summaries on this set of articles to study the inference-time behavior of the summarization models at different stages of their training trajectories.
2. **Token-level output probabilities for reference summaries**: Summarization models place a probability distribution over the entire output space and generated summaries are samples from the high probability regions. But looking only at these summaries does not tell us what *doesn't* get learned during training. To understand this aspect, we additionally analyze the models' output probabilities for reference summaries. Comparing reference summaries from low probability and high probability regions can provide further insight into the model behavior.

2.2 Case Study 1: Abstractiveness

Hypothesis Reference summaries from the three summarization datasets: XSUM, MEDIASUM and CNNDM exhibit varying degrees of abstraction. In this section, we aim to study the learning trajectory of this property during fine-tuning. We measure the degree of abstraction of the generated or reference summaries by the fraction of copied n-grams from the source article, for $n \in \{1, 2, 3, 4\}$, which we call n-gram overlap. **We hypothesize that a pre-trained model trained on some dataset should emulate its n-gram overlap statistics when evaluated on held-out instances from that dataset.**

Results Figure 1 shows the n-gram overlap of generated summaries (800 examples from the dev set) at different stages of the training process. The dotted lines in the graph represent the n-gram overlap of the reference summaries with the source article; this is the target degree of abstractiveness for the summarization models. The graphs show that for both BART- and PEGASUS-based models, the generated summaries exhibit high overlap at the start of the training process, probably because the model parameters are initialized with BART-LARGE or PEGASUS-LARGE which include high amount of copying. This overlap steadily decreases with more training steps.

However, the three summarization models show varying degrees of success at achieving the target level of abstractiveness in each dataset. For the XSUM dataset, the model behavior approaches the target abstractiveness quite early in the training process (after only 10% of the training for BART and approximately 30% of the training for PEGASUS), after which it plateaus. However, Figure 2 shows that the quality of the generated summaries continues to improve with more training: for BART, it increases from 41.9 ROUGE-1 at 10% of the

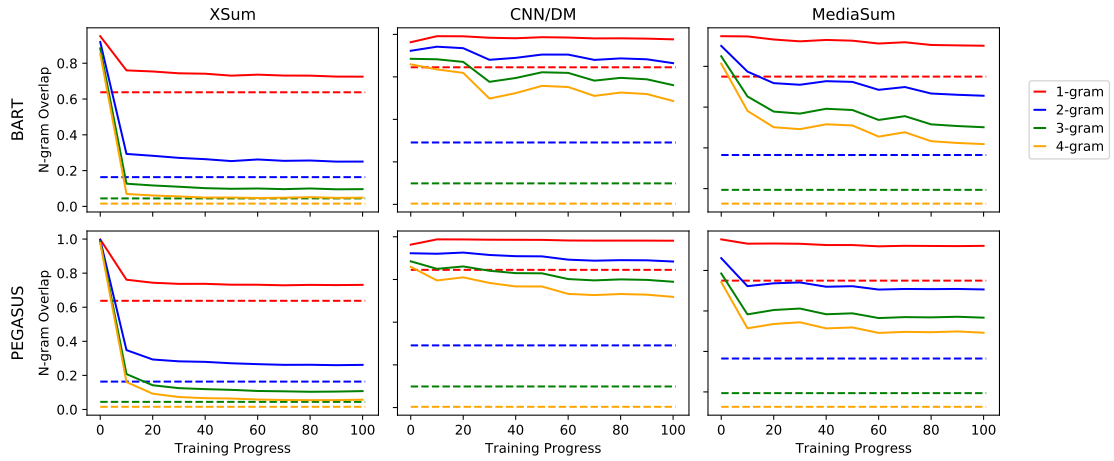

Figure 1: N-gram overlap of the generated summaries with the source article at different time steps. For CNNDM and MEDIASUM, the summaries fail to achieve the target degree of abstractiveness (denoted by the dotted lines).

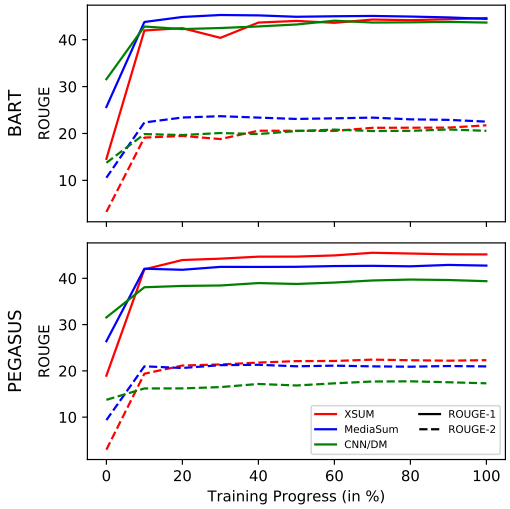

Figure 2: ROUGE scores of the generated summaries of all datasets at different training stages.

training, to 44.7 at the end of the training process. On the other hand, **for both CNNDM and MEDIASUM, the model generated summaries never achieve the target level of abstractiveness**. This is especially true for CNNDM; the n-gram overlap stabilizes after 30% of the training for BART and 60% for PEGASUS, differing substantially from the gold. For MEDIASUM, the the BART model shows a steadily decreasing trend, although it is not accompanied by a corresponding increase in quality (see Figure 2).

Interestingly, XSUM models shows greater success at achieving the target degree of abstraction compared to the others, even though their target abstractiveness is lower and involves a greater change in the model behavior from the initial stage.

Analysis Why do the other models generally fail to achieve the target n-gram overlap? Our hypothesis is that summarization models overfit on the *easier* examples in the training dataset, i.e. those that have high word level overlap with the source article, and this is exacerbated in CNNDM and MEDIASUM datasets which include a large fraction of such high overlap examples, compared to XSUM which has a lower fraction of these. Prior work has reported similar observations about overfiting for sequence classification tasks (Utama et al., 2020b).

To test this hypothesis, we randomly sample 1000 examples from the training data and compare the token-level output probabilities of high overlap (easier-to-learn) and low overlap (harder-to-learn) examples at different stages of the training process. These are chosen as the top and bottom 25% of the samples in terms of bigram overlap respectively. We conduct this analysis only for the BART-based models, shown in Figure 3. For the XSUM dataset, we observe that although the mean output probability of the high overlap summaries is generally higher, the model also assigns similarly high probabilities to the low overlap examples. On the other hand, for both CNNDM and MEDIASUM, there exists a substantial difference between the probabilities from these two sets of examples, resulting in the generation of more extractive summaries for these datasets.

Conclusions For both CNNDM and MEDIASUM, models do not achieve their target level of n-gram overlap. Although Figure 3 shows that the model performance on the low overlap, i.e. harder examples steadily improves as training progresses, **this does not translate into improvement in inference-time abstractiveness performance of generated summaries**. During inference, gener-

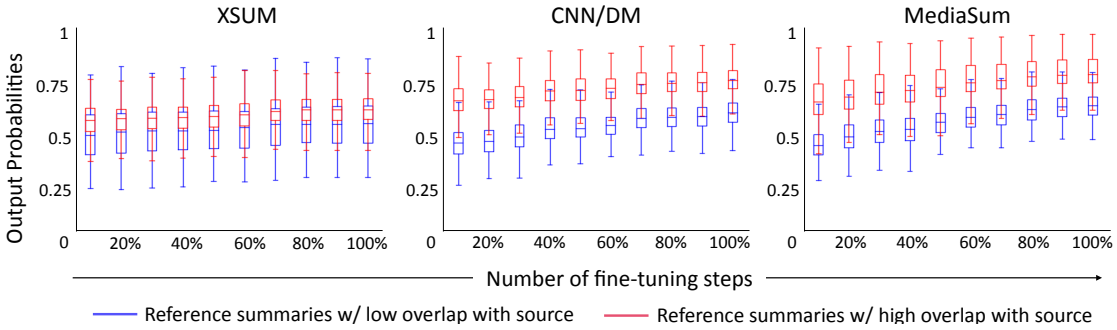

Figure 3: Comparison of summary-level output probabilities between high-overlap and low-overlap subsets for the BART models. For both CNNDM and MEDIASUM, high-overlap summaries are predicted with substantially higher confidence compared to low-overlap examples.

ated summaries are constructed by sampling the highest probability token at each time step (assume greedy decoding). Therefore, even though the probability of sampling abstractive tokens increases (blue boxes in Figure 3), it is still substantially lower than that of sampling extractive summaries (red boxes in Figure 3) and the model would prefer to generate more extractive summaries. In this respect, generation models differ from sequence classification models such as BERT-based models in how such graphs must be interpreted. For the latter, improvement in performance on harder examples indicate that the model would similarly perform better when it encounters these in the test data.

These dynamics indicate that training longer is insufficient to get better performance. Deeper modifications to the training procedure are needed to result in better test-time behavior.

2.3 Case Study 2: Factuality

Hypothesis Next, we evaluate the factual correctness of the generated summaries. Prior work (Maynez et al., 2020; Goyal and Durrett, 2021) has shown that BART-based summarization models, despite their impressive ROUGE scores, tend to produce highly non-factual summaries. In this section, we study how the factuality of generated summaries evolves during training. **We hypothesize that models start by making some factual errors in the initial stages. Longer training, however, should gradually lead to better factuality as they learn from the data, albeit never becoming perfect.**

Results To measure factuality, we use factuality models provided by Goyal and Durrett (2021). Given an input article A, and a generated summary S', the model predicts a factualilty label $y \in \{\text{non} - \text{factual}, \text{factual}\}$, denoting whether

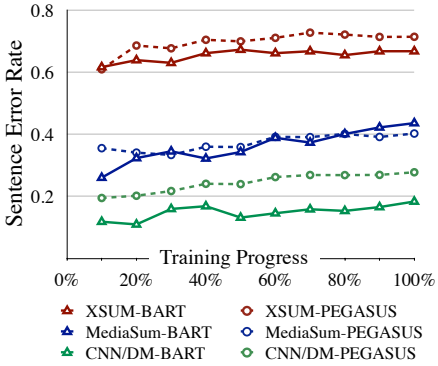

Figure 4: Factuality Sentence Error Rate of the generated summaries at different time steps during training.

the summary S' contains factual errors or not respectively. We directly use their pre-trained factuality models for XSUM and CNNDM. For the MEDIASUM dataset, we use the CNNDM model as its generated summaries are closer to the ones from MEDIASUM in terms of abstractiveness and length. We report the sentence error rate (SER) for 800 (article, generated summary) pairs from the development set at each training checkpoint. SER is computed as the fraction of generated sentences that are non-factual with respect to the article A.

Figure 4 shows the SER at different training steps for all BART- and PEGASUS-based models. First, we see that **the sentence-level error rate is roughly proportional to the abstractiveness of the generated summaries** for the three datasets: the generated summaries in XSUM have high error rates compared to the other datasets. Moreover, we see that the sentence-level error rate trajectories of both MEDIASUM and CNNDM mirrors the corresponding changes in abstractiveness in Figure 1. For instance, for the BART-based CNNDM model, the sudden drop in n-gram overlap at 30% is accompanied by a corresponding increase in the sentence-level error rates. Similarly, the error rate

steadily increases for the PEGASUS-based CNNDM model, following the steady decrease in overlap.

Apart from abstractiveness, recent work (Maynez et al., 2020; Goyal and Durrett, 2021) has identified the inherent noise in XSUM's reference summaries as a major reason for factuality errors. They show that around 70% of XSUM's training data consists of hallucinated content in gold summaries, which encourages the models to similarly learn to hallucinate facts. Figure 5 shows an illustrative example comparing the learning process of factual and hallucinated content in gold summaries during training. The graph plots the change in predicted probabilities for tokens in the reference summary. It shows that the model learns to predict *correct* information (packages) with high confidence early in the training process. On the other hand, hallucinated information is learnt mid-way through the training (after 40% progress). Moreover, throughout the training process, we notice that hallucinated tokens from the reference summaries are generally predicted with lower confidence than factual tokens. We use this observation to distinguish between factual and non-factual reference summaries in Section 3.3.

Conclusions For XSUM, as a model trains for longer, it learns idiosyncrasies and hallucinations in the training data. These do not systematically result in higher amounts of abstractiveness, but instead only yield a gradual increase in factual errors. Once again, training for longer is not the answer.

3 Improving Training

In Section 2.2, we saw evidence that summarization models tend to overfit on *easier* examples, i.e., the more extractive examples that are learnt earlier in the training process. On the other hand, Section 2.3 showed that for noisy datasets such as XSUM, correct information is assigned high probability scores earlier in the training process whereas hallucinated tokens are learnt with lower confidence. In this section, we operationalize these observations to improve the abstractiveness of generated summaries for CNNDM and MEDIASUM, and factuality of generated summaries for XSUM.

3.1 Loss Truncation

The core idea behind our approach is to modify the loss computation during the later stages of the training process, either disregarding high loss tokens to encourage factuality or low loss tokens to encourage abstractiveness.

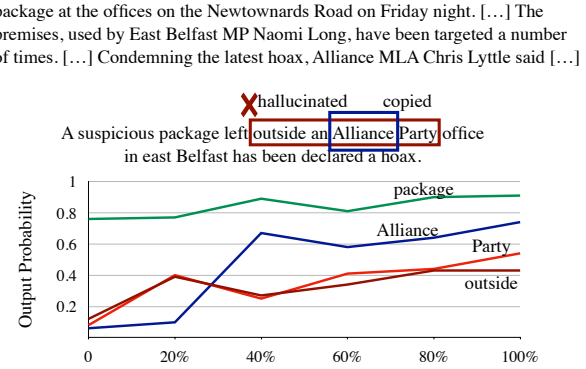

Figure 5: Example showing the output probabilities at different training stages for hallucinated and factual words in the gold summary. The graph shows that factual content is predicted with higher confidence.

Algorithm 1 LOSSTRUNCATION

Input: Model M, percentile p, standard training steps K, target $\in$ {abstractiveness, factuality}

for t in 0 to T
  $l_{0\ldots n} \leftarrow loss_M(x, s)$
  $q \leftarrow$ UpdateThresholdEstimate(l, p)
  if $t > K$
    if abstractiveness
      $m_j = \mathbb{1}[l_j > q]$ // truncate low loss tokens
    else if factuality
      $m_j = \mathbb{1}[l_j < q]$ // truncate high loss tokens
    $l_j \leftarrow m_j l_j$
  $M \leftarrow$ GradientUpdate(l)
return M

Algorithm 1 outlines our proposed approach. For the first K steps, standard training procedure is followed to train model M. After K steps, the loss function is modified to only incorporate the loss from a subset of tokens based on the summary property being targeted. To improve abstractiveness, tokens that have low loss ($l_j < q$) are excluded from the final loss computations; the assumption is that these are extractive tokens learnt with high confidence early in the training. Models trained using this strategy are denoted by ***+Abstractive***. On the other hand, tokens that have high loss ($l_j > q$) during the later stages of the training are excluded to encourage factuality. These models are denoted by ***+Factuality*** suffix. For both these different models, the threshold q between high and low loss is controlled through the percentile hyperparameter p. Throughout training, we dynamically update the loss statistics for the last 10k tokens, used to compute the top-p percentile threshold.

The overall loss truncation procedure is illustrated in Figure 7. For *+abstractive*, the losses associated with predicting the tokens ***left*** and ***an***

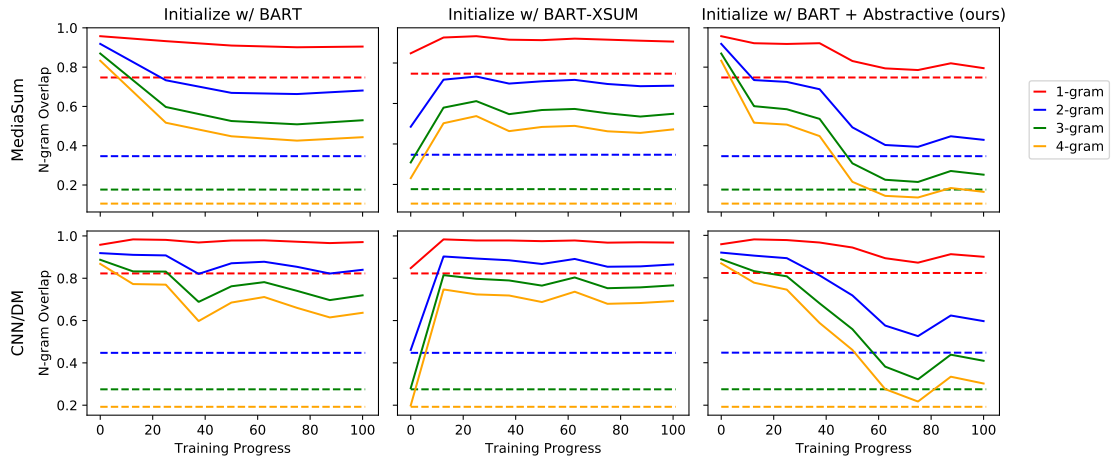

Figure 6: N-gram overlap of the generated summaries in CNNDM and MEDIASUM. Initializing from BART-XSUM offers no benefits over the baseline. On the other hand, loss truncation is successful at enforcing abstractiveness.

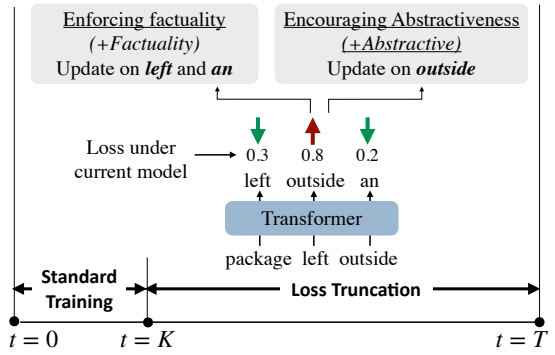

Figure 7: Modified training under loss truncation. After K steps of standard training, loss is computed on a subset of the tokens. To encourage factuality, high-loss tokens (↑) are excluded from the final loss computation whereas tokens with low loss (↓) are excluded to encourage abstractiveness.

are low, and hence removed from the final loss computation. For *+factuality*, the loss associated with token ***outside*** is high under the current model, and is excluded from the loss calculation.

Note that the loss truncation strategy to improve factuality is designed specifically to target the inherent noise in datasets like XSUM. Concretely, the approach attempts to identify and remove hallucinated content ***within*** gold summaries, enabling the model to only learn from factual content in the reference summaries. Therefore, datasets such as CNNDM and MEDIASUM are not the appropriate test bed for our factuality analysis as they do not suffer from similar noise in their training data.

3.2 Encouraging abstractiveness

First, we investigate the efficacy of the loss truncation approach at encouraging the abstractiveness of CNNDM or MEDIASUM models. We omit XSUM from our analysis of abstractiveness as the baseline BART model in Section 2 already achieves the target degree of abstraction for this dataset. Since both BART- and PEGASUS-based models have shown similar learning dynamics, we conduct experiments in this section only on the BART-based models.

Setup For both MEDIASUM and CNNDM, we train models for 8k steps. We set $K = 3$k: standard training is followed for the first 3k steps, followed by loss truncation for the remaining 5k steps. We set $p = 20$ for our experiments. For comparison, we include two baselines: (1) Model with parameters initialized with BART-LARGE (same as Section 2.2) and trained for 8k steps. (2) Model with parameters initialized with BART-LARGE-XSUM: its zero shot usage produces highly abstractive summaries. Here, we test if fine-tuning from this point helps with respect to abstractiveness.

Results Figure 6 shows the abstractiveness patterns for the different models for both CNNDM and MEDIASUM. For both datasets, while the models initialized with BART-LARGE-XSUM generate highly abstractive summaries in the beginning, fine-tuning for even a small number of steps results in overfitting to the extractive examples. In fact, the patterns for both the baselines look quite similar indicating that we do not derive any transfer learning benefits from the summarization skills encoded in BART-LARGE-XSUM. On the other hand, we see that the **model trained with loss truncation leads to substantially more abstractive summaries, across both datasets**. As expected, the level of abstractiveness drops sharply after 3k steps, i.e., when loss truncation is applied, and continues to decrease steadily. Moreover, the graphs

Input Article: Milan goalkeeper Dida has been cleared to play in next month's Champions League match at Shakhtar Donetsk after partially winning his appeal to UEFA against a two-match ban. Dida has had one game of his two-match ban suspended for a year following an appeal to UEFA. […] Dida sits out the home tie against Shakhtar on Wednesday after an inquiry …

Gold: Milan goalkeeper D ##ida **is partially successful against** a two-match UEFA ban …

	0 - 3k steps	> 3k steps	> 5k steps	> 7k steps
Milan	✓	✓	✗	✓
goalkeeper	✓	✗	✗	✗
D	✓	✗	✗	✗
##ida	✓	✗	✗	✗
partially	✓	✓	✓	✓
succesful	✓	✓	✓	✓
against	✓	✓	✓	✓

Figure 8: Example showing loss modification to improve abstractiveness. The table shows which tokens are retained (green checkmark) or dropped (red cross) from the loss computation at different training stages.

show that the models trained with loss truncation are able to come close to the target level of abstractiveness for the respective datasets, which both the baselines models struggled with. In Section 2.3, we discussed the trade-off between abstractiveness and factuality for summarization models. Our approach exposes a controllable lever, through the percentile hyperparameter p, that can be set by users to balance between these two properties based on their requirements.

Qualitative Analysis Figure 8 shows the loss modification for a training example at different stages using our *+Abstractive* strategy. The input article (truncated) and tokenized reference summary are stated at the top. Abstractive n-grams in the reference summary, i.e. those not exactly copied from the input article are highlighted in blue. The bottom half of the figure shows which tokens' prediction loss is included in the loss computation at different training stages. For the first 3k steps, all tokens' loss is aggregated. To encourage the model to learn abstractive strategies, we want to target the loss corresponding to the highlighted tokens. These represent an abstractive, somewhat subjective description of the events, and requires synthesizing information in a complex way. We observe that *+Abstractive* achieves this goal: the abstractive tokens (*partially, successful, against*) are high loss tokens after the initial training. Therefore, only these are included in the loss to train the model in subsequent time steps. On the other hand, tokens continuing a copied phrase (*goalkeeper*) usually have lower loss after the initial training and do not contribute to the gradient update in later stages.

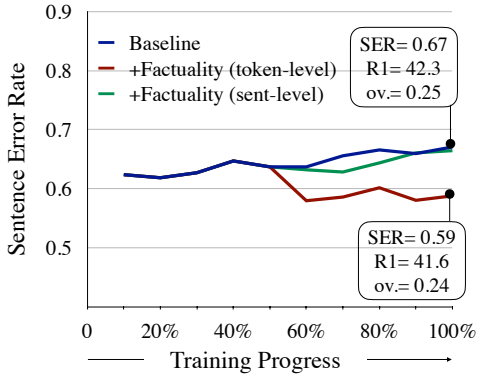

Figure 9: Factuality of output summaries for the baseline and loss truncation variants. The plot shows that token-level loss truncation improves factuality, with comparable results on abstractiveness and ROUGE.

3.3 Improving Factuality

Next, we study if similar down-weighting of knowledge learnt later in the training (*+Factuality*) can improve factual consistency of BART models. As mentioned previously, this strategy to improve factuality is designed for noisy datasets. Therefore, we only consider XSUM for our analysis.

Setup Apart from our token-level loss truncation outlined above, we also compare with a summary-level baseline from prior work (Kang and Hashimoto, 2020): summary-level loss is obtained during training (average of token-level losses) and those with loss greater than the p percentile mark are excluded from the loss computation. We call this *+Factuality sentence-level*. We set $p = 50$ for both our token- and sentence-level experiments. All models (including the baseline) are trained for a total of 10k steps: standard training for the first 5k steps, followed by loss truncation.

Results Figure 9 shows the factuality trajectory for the different models. We see that **the factual consistency of the generated summaries improves when token-level loss truncation is enforced**, dropping after 5k steps when the *+Factuality token-level* loss modification is applied. On the other hand, the summary-level approach from prior work does not lead to better factuality compared to the baseline. We hypothesize that this is because factual errors occur locally within the summary; 3-4 erroneous words within a 20 word summary. Therefore, averaging over all tokens makes it harder to distinguish between factual and non-factual summaries. Moreover, we also observe that **the token-level approach leads to better factuality without compromising on abstractiveness**.

Input Article: Lam, 28, joined the club in 2014, […] has ignored interest elsewhere to re-sign. […] "I feel I've got unfinished business here. […] I'm not getting any younger, two more years takes me up to 30 and then I'll have to start thinking about what I do after rugby. There are not too many years left in me and I'd like to see my years out at Bristol."

Gold: Bristol **flank ##er Jack** Lam has signed a new **two-year** contract with the **Championship** club **until 2018**.

	0 - 5k steps	> 5k steps	> 7k steps	> 9k steps
Bristol	✓	✗	✓	✓
flank	✓	✗	✗	✗
##er	✓	✓	✓	✓
Jack	✓	✗	✗	✗
Lam	✓	✓	✓	✓
two	✓	✗	✗	✗
year	✓	✓	✓	✓
Champ-ionship	✓	✗	✗	✗
club	✓	✓	✗	✓
until	✓	✗	✗	✗
2018	✓	✗	✗	✗

Figure 10: Example illustrating +*factuality* loss modification. The table shows which tokens are retained or dropped from the loss computation at each training stage. We can see that high-loss generally corresponds with hallucinated content.

Recent work (Ladhak et al., 2021) has shown that most prior work enforces factuality by sacrificing on the abstractiveness of generated summaries. Our analysis in Section 2.3 demonstrated a similar trade-off between factuality and abstractiveness. However, we see that our proposed loss truncation approach improves factuality without sacrificing the abstractiveness of generated summaries.

Qualitative Analysis Figure 10 shows an article-summary pair from XSUM training data. The hallucinated information in the reference summary, i.e. unsupported by the article, is highlighted in red. Claims that are similarly unsupported but stated in the article in other contexts are in blue. The correct parts of the gold summary are in black. The table at the bottom outlines which tokens' loss is included in the loss computation during training at different stages of the training, with high-loss (top-p percentile) tokens being *excluded*.

For the first 5k steps, losses corresponding to all tokens are aggregated. Thereafter, we see that the high-losses generally correspond with non-supported tokens and are removed. For e.g., the input article does not mention the first name *Jack* of player *Jack Lam*, and the loss corresponding to predicting *Jack* is removed from the overall loss. Similarly, other hallucinated tokens are successfully identified and removed, such as 'until 2018' and 'Championship'. However, some hallucinated tokens have low loss (and get retained in loss computation) if the probability of predicting them is high due to their prefix. For example, although *flank* is correctly identified as unsupported, the probability of predicting the ensuing subword *##er* is high (i.e. low loss). Similarly, although *two* is correctly identified as unsupported, the model predicts *year* with high confidence.

4 Related Work

Abstractive Summarization Prior work in abstractive summarization has evaluated summaries along various parameters such as grammaticality and informativeness (Woodsend and Lapata, 2012), agreement with reference (Lin, 2004; Zhao et al., 2019) and content selection (Nenkova and Passonneau, 2004; Deutsch et al., 2021). Recently, approaches to evaluate the factual correctness of abstractive summarization have been proposed (Falke et al., 2019; Kryscinski et al., 2020; Goyal and Durrett, 2020). However, all these have focused on only evaluating the final generated summary. Finally, both improving abstractiveness (Song et al., 2020) and factuality (Goyal and Durrett, 2021) have been explored in recent work; in this paper, we explore if simpler techniques inspired by the training dynamics can achieve similar goals.

Evaluating across time steps Recent work has studied learning dynamics of LSTM models (Saphra and Lopez, 2019) and pre-trained transformer models (Liu et al., 2021) across aspects such as linguistic knowledge, topicalization, reasoning, etc. Another line of work has explored this in the context of mitigating known dataset biases (Gururangan et al., 2018) for tasks such as paraphrase identification, entailment, etc. (He et al., 2019; Utama et al., 2020a). Broadly, these have proposed techniques such as example reweighting (Schuster et al., 2019), ensembling (Clark et al., 2019) or loss truncation (Kang and Hashimoto, 2020) to modify the model's learnt behavior.

5 Conclusion

In this paper, we study when different summarization *skills* are learnt during training. We show that copy behavior is learnt early while hallucination is learnt in the later stages. Based on these observations, we propose a simple token-level loss truncation strategy that can be used to achieve notable improvements in abstractiveness for CNNDM and MEDIASUM, and factuality in XSUM.

References

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. 2019. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. 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 4060–4073.

Daniel Deutsch, Tania Bedrax-Weiss, and Dan Roth. 2021. Towards question-answering as an automatic metric for evaluating the content quality of a summary. *Transactions of the Association for Computational Linguistics*, 9:774–789.

Tobias Falke, Leonardo FR 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.

Tanya Goyal and Greg Durrett. 2020. Evaluating factuality in generation with dependency-level entailment. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 3592–3603.

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.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A Smith. 2018. Annotation artifacts in natural language inference data. 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 107–112.

He He, Sheng Zha, and Haohan Wang. 2019. Unlearn dataset bias in natural language inference by fitting the residual. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 132–142.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in neural information processing systems*, pages 1693–1701.

Daniel Kang and Tatsunori Hashimoto. 2020. Improved natural language generation via loss truncation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 718–731.

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.

Faisal Ladhak, Esin Durmus, He He, Claire Cardie, and Kathleen McKeown. 2021. Faithful or extractive? on mitigating the faithfulness-abstractiveness tradeoff in abstractive summarization. *EMNLP*.

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*, 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.

Leo Z Liu, Yizhong Wang, Jungo Kasai, Hannaneh Hajishirzi, and Noah A Smith. 2021. Probing across time: What does roberta know and when? *arXiv preprint arXiv:2104.07885*.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan Thomas Mcdonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of The 58th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Caglar Gulcehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290.

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.

Ani Nenkova and Rebecca Passonneau. 2004. Evaluating content selection in summarization: The pyramid method. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 145–152, Boston, Massachusetts, USA. Association for Computational Linguistics.

Naomi Saphra and Adam Lopez. 2019. Understanding learning dynamics of language models with svcca. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,*

Volume 1 (Long and Short Papers), pages 3257–3267.

Tal Schuster, Darsh Shah, Yun Jie Serene Yeo, Daniel Roberto Filizzola Ortiz, Enrico Santus, and Regina Barzilay. 2019. Towards debiasing fact verification 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 3410–3416.

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, pages 8902–8909.

Prasetya Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. 2020a. Mind the trade-off: Debiasing nlu models without degrading the in-distribution performance. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8717–8729.

Prasetya Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. 2020b. Towards debiasing nlu models from unknown biases. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7597–7610.

Kristian Woodsend and Mirella Lapata. 2012. Multiple aspect summarization using integer linear programming. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 233–243.

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

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M Meyer, and Steffen Eger. 2019. Moverscore: Text generation evaluating with contextualized embeddings and earth mover distance. 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 563–578.

Chenguang Zhu, Yang Liu, Jie Mei, and Michael Zeng. 2021. Mediasum: A large-scale media interview dataset for dialogue summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5927–5934.

A Implementation Details

For experiments in Section 2, we train summarization models on the entire training data for XSUM, NPR subset for MEDIASUM and 50k randomly sampled examples from CNNDM (we found that this was enough to replicate the results of state of the art models). All our experiments are conducted using the Huggingface Library. Table 1 lists the hyperparameters used for fine-tuning the models and during inference.

For training	
Computing Infrastructure	32GB NVIDIA V100 GPU
Max Input Seq Length	1024
Max Output Seq Length	128
Optimizer	Adam
Optimizer Params	$\beta = (0.9; 0.999); \epsilon = 10^{-8}$
Learning Rate Decay	Linear
Learning rate	2e-5
Weight Decay	0
Warmup Steps	0
Max Gradient Norm	1
Batch size	16
For inference: XSUM	
Num beams	6
Length Penalty	2
No repetition size	3-grams
Min-Length	10
Max Length	60
For inference: CNNDM & MEDIASUM	
Num beams	5
Length Penalty	1
No repetition size	3-grams
Min-Length	20
Max Length	200

Table 1: Hyperparameters used for both the BART- and PEGASUS-based summarization models.

B Example Summaries

Table 2 provides examples of generated summaries obtained from the standard BART and BART *+Abstractive* models. The examples show that the latter lead to more abstractive summaries compared to the baseline. Table 3 compares generated summaries using the standard and *+Factuality* model aimed at improving factuality.

Input Article: Naypyidaw, Myanmar (CNN) Twenty-one people are dead and 21 missing after a ferry capsized in the Southeast Asia nation of Myanmar. Myanmar's Ministry of Information said in a statement that the ship capsized Friday night as it sailed, in bad weather conditions, around the city of Sittwe. That's when a large wave crashed into the ferry, causing it capsize near Myaybone and Myaukkyine islands. Authorities have managed to rescue at least 167 people, according to the information ministry for Myanmar, which is also known as Burma. Pictures from the government showed rescue workers helping people off a boat onto the land. Sittwe is the capital of Rakhine state and sits on the Bay of Bengal, about 55 miles (90 kilometers) from the Bangladesh border. This weekend's weather forecast for the city calls for some clouds giving way to clear skies, with high daytime temperatures expected to be in the 30s Celsius (80s to 90s Fahrenheit). Fatal ferry disasters are nothing new to the region. Last month, at least 68 people died when a packed double-decker ferry sank while on the Padma River north of neighboring Bangladesh's capital, Dhaka, officials said. A cargo vessel hit the ferry, causing it to overturn and trapping passengers on its lower deck. Forty-five people died in an accident on the same river in August. In May 2013, several boats carrying as many as 150 people were thought to have capsized near Myanmar's western coast ahead of a storm approaching the area. Those boats were carrying Rohingya, members of Myanmar's long-suffering Muslim minority, Thailand-based U.N. official Kirsten Mildren said at the time. Journalist Manny Muang reported from Myanmar, and CNN's Greg Botelho wrote this story from Atlanta.
Reference: 167 people have been rescued, Myanmar's government says. The ferry capsized after being hit by a large wave in bad weather conditions.
Baseline BART: The ship capsized Friday night as it sailed in bad weather conditions. Authorities have managed to rescue at least 167 people, according to the information ministry. Fatal ferry disasters are nothing new to the region.
BART *+Factuality*: At least 21 dead after ferry capsizes near Sittwe. At least 167 people have been rescued from boat. Fatal ferry accidents are nothing new to region.

Input Article: (CNN) NATO jets scrambled to intercept Russian military aircraft as they neared Latvian airspace, officials said on Wednesday. Estonian radar detected the aircraft over the Baltic Sea on Tuesday night, NATO said. Other than the lead aircraft, NATO said, none of the other Russian military aircraft was on a flight plan. NATO sent jets to identify the planes and later reported that the military aircraft flew on into Russian airspace. NATO didn't say how many Russian aircraft were involved. The flights come as Russia's Northern Fleet has been placed on full combat alert for military exercises involving nearly 40,000 troops and 50 warships. The exercises have rattled nerves in nearby NATO states, including Latvia, where U.S. troops and equipment recently arrived for NATO training, and where fears are growing about Russian President Vladimir Putin's next move. At the same time on Wednesday, Putin joined a celebration in Moscow's Red Square, where Russians celebrated the one-year anniversary of the annexation of Crimea. NATO has condemned the annexation as an illegal territory grab and is boosting its troop presence in the region in what officials say is an effort to discourage Putin from encroaching into other countries. Putin describes the annexation as a " reunification, " saying that Crimea's residents overwhelmingly voted to be part of Russia. CNN's Don Melvin and Catherine E. Shoichet contributed to this report.
Reference: Russian military aircraft are intercepted by NATO jets. NATO says the military aircraft weren't on a flight plan. Russia is conducting military exercises.
Baseline BART: NATO jets scrambled to intercept Russian military aircraft as they neared Latvian airspace. Russian Northern Fleet has been placed on full combat alert for military exercises.
BART *+Factuality*: Russian military planes flew into Latvian airspace, according to NATO. Flights are part of Russia's preparations for major military exercises involving 40,000 troops, 50 warships.

Table 2: Generated summaries from CNNDM dataset using the baseline BART model and the BART *+Abstractive* model proposed in this work. Longer copies phrases/sentences are underlined. Examples show that the generated summaries of the *+Abstractive* model are much more abstractive compared to the baseline.

Input Article: Visitors will be shown updates from authorities, news articles, emergency telephone numbers and other useful information in a single place. The SOS Alerts facility can also be set to trigger mobile notifications to those nearby to affected locations. However, Google is still seeking partners to improve the service. The initiative builds on earlier emergency response efforts from the US firm, including its Person Finder and Crisis Map tools. But this time, rather than requiring users to go to special sections of its site, SOS Alerts attempts to bring key information about incidents directly into two of Google's most used services. When activated, the Maps tool reveals, among other things, areas that should be avoided, which roads have been closed and places users can seek refuge. Data gathered from the firm's crowdsourced Waze mapping platform also makes it possible to see where traffic jams, accidents and other problems have been reported by the public. The level of detail shown within the Search tool depends on whether the person carrying out the query is close to the incident. If nearby, they are presented with links to official alerts, tweets from first responders, and useful short phrases in the local language. Those searching from afar are shown less detail unless they click for more information, but they may also be told how to make donations to charities involved in clean-up operations, if Google believes it to be appropriate. "In situations of crisis, the need for information is crucial," Yossi Matias, the firm's vice-president of engineering, told the BBC. "People need to know what's going on - anything that may be related to their safety, or any action they should be taking." He added that Google had set up a dedicated team to decide which events warranted an SOS Alert, but declined to reveal how many people had been assigned to it. Facebook - which offers a parallel service to let members in the vicinity of a disaster tell friends they are safe - has at times been criticised for activating it under "inappropriate" circumstances. Google has joined forces with government bodies, the Red Cross and various weather-forecasting organisations to help provide SOS Alerts in 12 countries. They include local organisations in the US, Japan, the Philippines, Australia and Canada. But it has yet to secure partners in the UK and other European nations. SOS Alerts will still cover events there, but will contain less information as a consequence until information-sharing arrangements are struck. "In times of crisis, more and more people are turning to online sources of information to find out what to do," Omar Abou-Samra from the International Federation of Red Cross told the BBC. "Designed to be shared in tandem with public alerts, the service provides localised lifesaving information that people can immediately act on to protect themselves and their families."
Reference: Google has begun rounding up information about unfolding natural disasters, terrorism and other crises within its Search and Maps tools.
Baseline BART: Google is to expand its SOS Alerts service to include information about natural disasters and other major events on its home page.
BART +*Factuality* Google has launched a new service to help users nearby by bringing key information about disasters to its Maps and Search tools.

Input Article: The country's consumer watchdog has taken Apple to court for false advertising because the tablet computer does not work on Australia's 4G network. Apple's lawyers said they were willing to publish a clarification. However the company does not accept that it misled customers. The Australian Competition and Consumer Commission (ACCC) said on Tuesday: "Apple's recent promotion of the new 'iPad with wi-fi + 4G' is misleading because it represents to Australian consumers that the product can, with a sim card, connect to a 4G mobile data network in Australia, when this is not the case." The watchdog then lodged a complaint at the Federal Court in Melbourne. At a preliminary hearing, Apple lawyer Paul Anastassiou said Apple had never claimed the device would work fully on the current 4G network operated by Telstra. Apple says the new iPad works on what is globally accepted to be a 4G network. The matter will go to a full trial on 2 May. The Apple iPad's third version went on sale earlier this month, with Australia the first country where it was available. Shoppers lined up by the hundreds at Apple stores on opening day and the company said it had been its strongest iPad launch to date. The ACCC said it was seeking an injunction on sales as well as a financial penalty against Apple, corrective advertising and refunds to consumers. On its website, Apple does state that 4G LTE is only supported on selected networks in the US and Canada.
Reference: US technology firm Apple has offered to refund Australian customers who felt misled about the 4G capabilities of the new iPad.
Baseline BART: Australia is the first country where the new iPad does not work on a 4G network.
BART +*Factuality*: Apple has been accused of misleading Australians about the new iPad.

Table 3: Comparison of summaries generated by the standard BART model and a BART +*Factuality* model trained using our proposed loss truncation strategy. The errors made by the models are highlighted in red.