Abstract

In this work, we propose a method for incorporating question-answering (QA) signals into a summarization model. Our method identifies salient noun phrases (NPs) in the input document by automatically generating wh-questions that are answered by the NPs and automatically determining whether those questions are answered in the gold summaries. This QA-based signal is incorporated into a two-stage summarization model which first marks salient NPs in the input document using a classification model, then conditionally generates a summary. Our experiments demonstrate that the models trained using QA-based supervision generate higher-quality summaries than baseline methods of identifying salient spans on benchmark summarization datasets. Further, we show that the content of the generated summaries can be controlled based on which NPs are marked in the input document. Finally, we propose a method of augmenting the training data so the gold summaries are more consistent with the marked input spans used during training and show how this results in models which learn to better exclude unmarked document content.

1 Introduction

Abstractive sequence-to-sequence summarization models have become very effective methods of easily generating summaries of input documents (Rush et al., 2015; Nallapati et al., 2016; Lewis et al., 2020).

Previous work has demonstrated that conditioning the summary generation on salient document sentences results in higher-quality summaries and more controllable summarization models (Chen and Bansal, 2018; Dou et al., 2021). Salient sentences are typically identified during training by lexical overlap with the gold summaries (Nallapati et al., 2017) and predicted during inference.

Although marking different sentences as salient allows for some controllability over the content of the summary, desired summary content cannot be specified at the sub-sentence level. Further, labeling sentences as salient via n-gram overlap does not directly take the predicate-argument structure of the text into account, which could result in a lower-quality supervision signal that misidentifies which particular instance of an n-gram is salient.

In this work, we propose to condition the summary generation on salient sub-sentence level spans which are identified by reasoning about the predicate-argument relations in the text.

We mark noun phrases (NPs) in the input document as salient if the predicate-argument relation they participate in is present in the gold summary (§2). This idea is implemented using automatic question generation (QG) and answering (QA). For each NP, a wh-question that is answered by the
NP is generated from the text. Then, the NP is marked as salient if the corresponding wh-question is correctly answered in the gold summary according to an automatic QA model, resulting in a more precise, sub-sentence level supervision signal (see Fig. 1).

The QA-based salience signal is incorporated into a two-stage summarization model (§3). First, a phrase salience classifier is trained to identify which NPs in the document are salient. Then, the predicted salient spans are marked in the input document with special tokens and used to conditionally generate a summary of the document with a fine-tuned BART model (Lewis et al., 2020).

Our experiments on three different summarization datasets show that the two-stage model trained with QA-based salient span supervision generates higher-quality summaries than lexical baseline methods of identifying salient spans on more extractive datasets according to several automatic evaluation metrics (§6.2). Further, we demonstrate how the summary content can be controlled via marking the input spans by showing that the QA-based model generates summaries that answer the wh-questions corresponding to the marked NPs far more frequently and concisely than BART does (§6.3).

However, because our QA annotation method may not perfectly identify all of the gold summary content in the document (due to noise or unfaithful gold summaries), the generation model learns to include content which was not marked in the input document, an undesirable property of a controllable summarization model. Therefore, we propose a data augmentation procedure that removes sentences unsupported by the marked spans as well as generates new training examples that teach the model to generate a subset of the gold summaries sentences conditioned the corresponding salient spans with wh-questions that are answered by those sentences (§4). The model trained on the augmented data generates summaries that are 22% shorter with only a small reduction in the percent of answered questions, demonstrating how it successfully eliminates undesired summary content (§6.3).

The contributions of our work include: (1) a novel method of including QA-based signals into summarization generation; (2) a two-stage model for incorporating phrase-level supervision into a summarization system; and (3) a data-augmentation procedure which results in more controllable summarization models.

2 Question-Based Salience

We begin by describing how we use QA to identify salient spans of text in the input document and discuss the advantages of this approach.

We define a document NP as salient if its corresponding predicate-argument relation also appears in the gold summary. To identify such NPs automatically, we employ question-generation and question-answering models as follows.

For each NP in the source document, we use the sentence it appears in to automatically generate a wh-question for which the NP is the answer. This question-answer pair represents the predicate-argument relation that the NP participates in. Then, we assume if a second text can be used to correctly answer that question, it contains the same predicate-argument relation. Thus, we use a QA model to automatically answer the question against the gold summary and mark the NP as salient if the QA model predicts the question is answerable and the predicted answer is correct. In practice, we assume a predicted answer is correct if it shares at least one token in common with the NP which was used to generate the question.

An example of this procedure is illustrated in Fig. 2 for two occurrences of the NP “Sierra Leone.” Questions for each phrase are automatically generated from the input document and answered against
the gold summary. Since the QA model correctly answered the first question but predicted the second question is not answerable, only the first occurrence of “Sierra Leone” is marked as salient.

Specific implementation details of the generation and answering models can be found in §5.

2.1 Advantages of a QA-Based Approach

Using QA to identify salient spans of text has several advantages. First, because our QA approach operates at the phrase-level, it is able to be more precise about what specifically is salient in the document in contrast to sentence-level approaches. For example, in the second sentence of Fig. 1, the QA-based salience signal identifies “Jonathan” and “his defeat” as salient but not “written statement.” A sentence-level approach would mark the entire sentence as salient and thus cannot make that distinction.

Second, because the QA-based approach reasons about the predicate-argument structure of the text, it is able to distinguish between which specific instances of the same NP are salient and which are not. This is illustrated in Fig. 2 in which the first occurrence of “Sierra Leone” is marked as salient but the second is not because the gold summary does say the health care worker was infected in Sierra Leone, but it does not say it is one of the hardest hit countries. A salience signal that uses a bag-of-n-grams approach (e.g., ROUGE-based methods) cannot easily decide which instance “Sierra Leone” is salient.

If a summarization model was trained to use a salience signal that is based on the predicate-argument structure of the text — such as the QA-based approach here — these advantages would allow for more control over what content is included in the summary. Specific occurrences of NPs could be marked as salient, either manually or automatically, signaling to the summarization model that the corresponding predicate-argument relation should be included in the generated summary. This would not be possible with a sentence-level model (because it would lack the ability to mark individual NPs) or a phrase-level model which does not reason about the predicate-argument structure (because multiple instances of the same phrase would be indistinguishable, so it would not be clear which one was salient).

Although we argue a salience signal that reasons about the predicate-argument structure is necessary for a more controllable model, it is not a sufficient condition. For instance, if the salience signal is noisy, the model may not learn to always include the salient content in the generated summary, thus decreasing its controllability.

3 A Two-Stage, Span-Based Model

Next, we propose a two-stage, span-based model called SPANSUM that can incorporate the QA-based salience signals into the learning procedure. The first of the two stages, the span selection component, classifies salient spans within the text. The second stage, the generation component, generates the summary given the document and the salient spans. The details of each component are detailed next.

3.1 Salient Span Classifier

Given an input document \( d = [x_1, \ldots, x_n] \) and a set of spans \( S \), in which each span \( s_{i,j} \) represents a sequence of tokens \( x_{i}, \ldots, x_{j} \) in \( d \), the span classifier outputs a score for each span based on how salient it is in the document. Our definition of salience is discussed in §2.

Concretely, the input tokens are first encoded using BART. Then, the representation of a span is created by concatenating the BART encodings of the first and last tokens in the span. Finally, a linear classifier is trained using this encoding to predict the salience of each span.

A set of oracle spans \( S^* \subseteq S \) is used to train the model using a binary cross-entropy loss. When using the QA-based approach, \( S \) is the set of NPs in the document and \( S^* \) is the subset that our question generation/answering algorithm identified as salient. We reweight the loss term of each span such that positive and negative spans contribute equally. During inference, a score is predicted for each span in \( S \) and the top-\( k \) sorted by highest score are passed to the generation component. We choose the \( k \) spans independently, although they could also be selected jointly.

3.2 Generation Component

Given an input document and set of salient spans, the generation component produces a summary of the document. The salient spans are represented by inserting special tokens directly into the document’s sequence of tokens before and after the spans. For example, if span \( s_{4,5} \) was marked as
Usain Bolt rounded off the world championships Sunday by claiming his third gold in Moscow as he anchored Jamaica to victory in the men's 4x100m relay. The fastest man in the world charged clear of United…

Usain Bolt rounded off the world championships Sunday by claiming his third gold in Moscow as he anchored Jamaica to victory in the men's 4x100m relay. The fastest man in the world charged clear of United…

Figure 3: An example of our data augmentation procedure. The colors represent the mapping between document and summary spans. In this example, no span maps to the third summary sentence, so it is removed entirely. Then, new training instances are generated using the first summary sentence and first two summary sentences with their corresponding salient document spans.

salient, the document’s tokens would be represented as

... x3 [SS] x4 x5 [SE] x6 ...

where [SS] and [SE] mark the start and end of the span.

Since the salient spans are represented in the document tokens, we are able to directly train a sequence-to-sequence model to generate the gold summary from the modified document representation without any changes to the model’s architecture.

During training, we use oracle spans and the ground-truth summary to fine-tune BART using a standard cross-entropy loss function. For inference, the predicted salient spans from the span classifier are used instead of the oracle spans.

Although there is nothing to directly force the generation model to learn to include content based on the supervision provided by the salient spans, if the supervision is of high enough quality, we expect the model will learn to do so. As we show in §6.3, this enables controlling the content of the summary by marking spans in the input document.

4 Improving Controllability via Data Augmentation

In addition to learning that specific content should be included in the summary, a controllable generation model should also learn to not include undesired information. Our generation model may learn to include extra information for at least two reasons.

First, the gold summaries may include content which cannot be generated based on only the oracle salient spans that were used to train the generation model, so it may learn to output extra, unmarked information. This could happen if the question generation and answering models are imperfect (resulting in a noisy oracle) or if the gold summary contains information that cannot be mapped to the document. Second, if the model is trained to generate summaries of a certain length and the length of the summary necessary to include all of the information marked by the spans is smaller than those used for training — for example, because the number of marked spans is small — the model could generate additional information simply to increase the summary length.

An artifact of our oracle span annotation procedure enables us to address this controllability issue. If a document span is marked as salient, that means it has a corresponding phrase in the gold summary which expresses the same content. Therefore, the question generation and answering procedure creates a mapping between which parts of the gold summary should be able to be generated by marking different parts of the input document.

We propose to leverage this mapping to augment the training data in two ways. First, we remove any gold summary sentence which does not have a mapping to any part in the document. These sentences would encourage the model to generate additional content based on unmarked spans.

An example of these augmentations is included in Fig. 3.
5 Experimental Setup

Datasets Our experiments use three popular English single-document summarization datasets: CNN/DailyMail (Nallapati et al., 2016), XSum (Narayan et al., 2018), and NYTimes (Sandhaus, 2008). Specific details on the sizes of the datasets can be found in Appendix A.

Baselines & Other Work We compare the salient spans selected by our QA-based method against three baseline span selection methods. The first marks salient sentences by greedily selecting \( k \) sentences that maximize the ROUGE-2 score calculated against the gold summary, a popular method that is frequently used to train extractive summarization models (Nallapati et al., 2017) as well as other two-step abstractive systems (Chen and Bansal, 2018; Dou et al., 2021). The other two mark entities and NPs as salient if they appear in the gold summaries as determined by lexical matching. We only mark the first occurrence of the phrases as salient since we found that worked better than marking all occurrences.

Additionally, we compare our results to BART (the original implementation and our own; Lewis et al., 2020) since our models are built on top of BART. We also compare to GSum (Dou et al., 2021), which uses salient sentence guidance that is similar to our baseline salient sentence method. GSum encodes the additional guidance signal separately from the input document and uses the document and guidance encodings to generate the summary.

Summarization Evaluation Metrics The models’ summaries are automatically evaluated using three metrics which calculate a similarity score between the generated and gold summaries. ROUGE (Lin, 2004) compares the two summaries based on their lexical overlap. BERTScore (Zhang et al., 2020) calculates a similarity score between the summaries based on their tokens’ BERT embeddings (Devlin et al., 2019). QAEval (Deutsch et al., 2021) is a question-based evaluation metric, which generates questions from the gold summaries and answers them against the generated summaries. Its similarity score is equal to the average token F\(_1\) score calculated between the predicted and expected answers.

We additionally perform a human evaluation of summary quality on Mechanical Turk. We ask 3 Turkers to rate the quality of 50 summaries per model from the CNN/DailyMail dataset on a scale from 1 to 5 based on the importance of the information, faithfulness, fluency, and coherence. Details on the manual evaluation can be found in Appendix D.

Controllability Evaluation Metrics The controllability of our model is evaluated using the question recall. Given \( k \) marked spans, we define the question recall to be equal to the percent of the corresponding \( k \) wh-questions that are answered by the summary according to the question answering model. This approximates the recall on the desired predicate-argument structures in the summary. We additionally report the ratio between \( k \) and the length of the generated summary in tokens to measure the precision of the generated information. A larger value means the summary is more concise.

Implementation Details The question generation and answering models are the same as used by QAEval. The generation model is initialized with BART-Large and fine-tuned on data collected by Demszky et al. (2018). The answering model is initialized with ELECTRA-Large (Clark et al., 2020) and fine-tuned on SQuAD 2.0 (Rajpurkar et al., 2018).

The span classification and generation models are both initialized with BART-Large and fine-tuned on the respective datasets. They were trained for three and five epochs, respectively, and the model with the best precision@1 and ROUGE-2 F\(_1\), respectively, on the validation set were selected as the final models. See Appendix B for more specific implementation details.

6 Results

6.1 Salient Span Classifier Evaluation Fig. 4 contains the precision@\( k \) and recall@\( k \) of the span based classifiers. The “x” symbols denote the operating points used in the end-to-end model, which were chosen based on the number of spans that resulted in the highest ROUGE-1 F\(_1\) score on the validation data.

Interestingly, the operating points have very different precisions and recalls across the different methods, although we believe directly comparing their values could be misleading. For example, misidentifying a salient sentence is far more costly in terms of precision than incorrectly marking a NP as salient since the number of output sentences
is much less than the number of NPs. Further, because the generation models were trained with oracle spans, there could be some complex interaction between the two models in which the generation model may need a minimum level of span classification quality in order to generate a good summary. This minimum level could be different for each span type.

6.2 Generation & End-to-End Evaluation

Automatic Evaluation Table 1 contains the models’ performances as evaluated by automatic metrics. The top section contains the results for BART and GSum. The middle contains SPANSUM scores assuming the oracle spans are given to the generation model, whereas the bottom section contains the end-to-end SPANSUM results which use the predicted spans.

On CNN/DailyMail and NYTimes, the end-to-end QA-based model performs the best on all metrics across the different span labeling methods, whereas on XSum, it is statistically tied or worse than using salient sentences. On NYTimes, the QA model is the best across all models, including baselines and other work. Although the sentence-based model may sometimes do better on XSum, both methods perform worse than our implementation of BART on XSum. Their respective oracle scores are not much higher than BART, suggesting the generation model did not find the salient span supervision particularly helpful.

One explanation for this result is that CNN/DailyMail and NYTimes are datasets for which extractive approaches do well, in contrast to XSum which is true for abstractive approaches. Since all of our span labeling techniques rely on the input documents directly stating the content in the summary, it is not surprising that our models do better on the more extractive datasets. This is consistent with results reported by Dou et al. (2021) for GSum.

Interestingly, the span labeling method with the highest oracle performance on all three datasets is the lexical NP matching baseline. Lexically matching NPs likely has higher recall than our QA-based approach because the question generation and answering models are imperfect, so some phrases are missed. Further, because news articles have a strong lead bias, marking the first occurrence of a NP as salient (instead of a later one) likely has relatively high precision. The QA model’s advantage of being able to reason about the predicate–argument structure of the text is smaller when the lead bias is strong. However, it seems that because the NP end-to-end performance is worse than the QA-based model, the salient phrase classifier trained with lexical NP supervision must not be able to provide the generation model with a high-enough quality signal to generate a good summary.

In comparison to GSum (which uses salient sentence guidance), the QA model is slightly worse on CNN/DailyMail but better on NYTimes. Regardless of which model has higher automatic scores, our results show that it is possible to achieve equal or better performance with supervision that reasons about the semantics of the text rather than lexical matching, which was not true for Dou et al. (2021).

Human Evaluation Table 2 contains the results of evaluating BART and the sentence- and QA-based models on CNN/DailyMail using human summary quality annotations from Mechanical Turk. On average, our span-based methods have higher quality summaries than the baseline method of BART. After collecting annotations for 50 summaries on CNN/DailyMail, we were unable to obtain statistical significance between the two span-based models, however, doing so may be prohibitively difficult (Wei and Jia, 2021).

6.3 Controllability Evaluation

Automatic Evaluation The controllability of our QA-based model is evaluated in Fig. 5. We plot the question recall and the ratio between \( k \) and the length of the generated summaries for the top \( k \) most salient spans output by the QA-based
Table 1: The automatic metric results for the baselines and other work (top), models that use oracle spans (middle), and end-to-end models (bottom) evaluated with ROUGE (R1, R2, RL), BERTScore (BSc), and QAEval (QAE). Values in bold are statistically the best in each section and † marks the best values overall (excluding oracle) using a permutation test with $\alpha = 0.05$.

| Method              | CNN/DailyMail | XSum | NYTtimes |
|---------------------|---------------|------|----------|
| **Baselines & Other Work** |                |      |          |
| BART                | 44.2          | 21.3 | 40.9     |
| BART (ours)         | 44.1          | 21.0 | 40.9     |
| GSUM                | 46.0†         | 22.4 | 37.2†    |

**SPANSUM – Oracle Spans**

| Metric      | CNN/DailyMail | XSum | NYTtimes |
|-------------|---------------|------|----------|
| Sentences   | 51.7          | 29.9 | 48.8     |
| Entities    | 51.5          | 27.6 | 48.0     |
| NPs         | 59.6          | 34.6 | 55.8     |
| QAs         | 55.3          | 31.4 | 51.9     |

**SPANSUM – End-to-End**

| Metric      | CNN/DailyMail | XSum | NYTtimes |
|-------------|---------------|------|----------|
| Sentences   | 45.0          | 21.8 | 41.8     |
| Entities    | 43.5          | 20.3 | 40.4     |
| NPs         | 44.8          | 21.0 | 41.6     |
| QAs         | 45.5          | 21.9 | 42.4     |

Table 2: Summary quality scores according to humans. Results in bold are statistically tied for the best score.

| Method | Sentences | QA |
|--------|-----------|----|
| BART   | 3.76      | 3.86 | 4.00 |

Although BART’s question recall is initially higher than the QA models’ recalls, as $k$ increases it falls lower. We suspect this is because BART has learned to include the same content that the span classifier also identifies as salient when $k$ is small and the length of its summaries allows it to cover more content. However, when $k$ increases, the span classifier potentially predicts different spans as salient than what BART learned, resulting in divergent content and a lower recall for BART. The higher recall of the QA models demonstrates that their summary content is indeed being controlled via the input spans. Further, the QA models have far better $k$-to-length ratios, meaning their summaries are shorter than BART’s even when their recalls are higher, suggesting they generate far less content which is unrelated to the marked spans.

Among the QA-based models, we do observe a salient span classifier for various values of $k$ on CNN/DailyMail. These values are computed for the QA-based model trained on the original data as well as the augmented data described in §4, including only removing sentences that do not answer a question (“+Rm Sents”) plus also generating new training examples (“+New Examples”). We also include the results for BART (for which the summary is constant for all $k$) for relative comparisons.

![Figure 5: The percent of questions which correspond to the marked spans answered by the generated summaries (top) and the summary lengths in tokens (bottom). The QA-based methods have higher question recall than BART and are far more concise, demonstrating that marking input spans controls the summary content.](image-url)
Table 3: The percent of times the question corresponding to the $k$th occurrence of the most salient span according to the QA-based classifier was answered by the summary. The QA-based model is better able to generate non-lead document content.

| $k$ | 1 | 2 | 3 | 4 | 5 |
|-----|---|---|---|---|---|
| NPs | 69.3 | 59.0 | 40.2 | 33.1 | 29.4 |
| QAs | 75.0 | 64.2 | 45.9 | 41.2 | 34.2 |

small drop in recall when the model is trained with data augmentation. However, the data-augmented summaries express that information far more concisely (because the ratio between $k$ and the summary length is higher). For example, when 10 input spans are marked, there is a relative 0.9% and 3.2% drop in recall for removing sentences and the full data augmentation procedure, respectively, but the summary lengths are 14% and 22% shorter. Therefore, the data augmentation procedures do result in models which have learned to not generate extra content.

Controllability Example  Example summaries from the QA models and sentence-based model with different marked input spans are shown in Fig. 6. Because the sentence-based model is limited to marking full sentences, the content which is taken from the marked sentence cannot be further controlled. In contrast, the figure shows how the QA-based models’ summaries can be altered by marking different NPs within the sentence, thus demonstrating the benefits of phrase-level controllability.

The example in Fig. 6 also shows how the QA model trained on the augmented data improves controllability. The phrases which the standard model includes but the augmented model does not are marked in bold. The augmented model does a better job at excluding content which was not marked in the input document.

Examining Lead Bias  A desirable property of a controllable summarization model is that content at any position in the document can be selected to be included in the summary. Since the salient QA-based spans are selected by reasoning about the predicate-argument structure of the text, it should better achieve this property than the baseline lexical NP-based model, which only saw the first occurrence of a NP marked as salient during training. However, because there is a strong lead bias, choosing the first occurrence might often be the correct choice, and noise in the oracle QA annotations may contribute to the generation model ignoring salient spans in the middle or end of the document.

To evaluate how well the models achieved this property, we marked the $k$th occurrence of the most salient phrase output by the QA-based span classifier on CNN/DailyMail and calculated the question recall for the QA- and NP-based models in Table 3. Both models were trained using augmented training data (an equivalent augmentation procedure to the one described in §4 was used for the lexical NPs).

For each value of $k$, the QA model has higher question recall by around 5%, implying it can better select non-lead content and is thus more controllable. This result is unlikely explained by the lengths of the summaries, which are longer for the QA model by only 1 token on average (10 versus 11). As $k$ increases, the question recall does decrease significantly for both models, suggesting the models do begin to ignore spans toward the middle and end of the document.

7 Related Work

QA-Based Signals  QA-based signals have been used for evaluating summaries (Eyal et al., 2019; Durmus et al., 2020; Wang et al., 2020; Deutsch et al., 2021), including Scialom et al. (2021), who explore a similar notion of document salience. They have also been used to align content across documents (Weiss et al., 2021) as well as train summarization models (Arumae and Liu, 2018, 2019; Scialom et al., 2019). The models which incorporate QA-based signals typically do so using reinforcement learning. In contrast, our approach is simpler. We incorporate the QA-based signal by marking spans in the text, and our models are trained using easier-to-optimize cross-entropy objective functions.

Incorporating Additional Supervision  Recent work by Dou et al. (2021) proposes a framework for incorporating additional guidance into summarization models, called GSum. They separately encode the input document and the supervision signal, whereas we directly mark spans in the text. This allows for our generation component to have a simpler architecture than theirs. While they are able to encode any natural language string, our model provides more direct supervision by identifying which specific tokens are salient.

Other work has included predicate-argument
structure into summarization with the goal of producing more faithful summaries (Cao et al., 2018; Jin et al., 2020; Zhu et al., 2021). They represent the predicate-arguments either using dependency trees or OpenIE tuples, whereas we represent them via QA pairs. The key difference is that they are including that information to try and generate faithful summaries, whereas we are using it to supervise the models and predict which predicate-arguments are salient.

**Controllable Summarization** Work on controllable summarization has focused on aspects such as the length of the summary (Fan et al., 2018) and the content in an interactive setting (Shapira et al., 2017) or via prompting (He et al., 2020). Incorporating our QA-based signal via prompting may be difficult given the number of questions which would need to be concatenated onto the input.

Other approaches control content via planning as in entity templates (Narayan et al., 2021) or marking records in a data-to-text approach (Puduppully et al., 2019). The marked salient spans in our work could be viewed as a content plan as well.

**Data Augmentation** Previous work has proposed methods for removing sentences or full summaries from the training data in order to discourage the summarization model from learning to generate unfaithful information (Matsumaru et al., 2020; Nan et al., 2021; Narayan et al., 2021). In addition to removing sentences, we generate new training instances in order to learn to exclude content which is not marked as salient in the input, resulting in more controllable models.

**8 Conclusion**

In this work, we proposed a method for incorporating QA-based signals into a summarization model by automatically marking document NPs as salient based on whether a NP’s corresponding wh-question is answered correctly in the summary. We showed that incorporating this signal into our two-stage summarization model results in higher quality and more controllable summaries as compared to baseline methods of identifying salient spans. Finally, we demonstrated that our data augmentation algorithm, which attempts to ensure the span supervision is consistent with the gold summaries, improves controllability by eliminating unmarked content from the output summaries.

**References**

Kristjan Arumae and Fei Liu. 2018. Reinforced Extractive Summarization with Question-Focused Rewards. In Proceedings of ACL 2018, Student Research Workshop, pages 105–111, Melbourne, Australia. Association for Computational Linguistics.

Kristjan Arumae and Fei Liu. 2019. Guiding Extractive Summarization with Question-Answering Rewards. In Proceedings of the 2019 Conference of the North American Chapter of the Association for
Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2018. Faithful to the Original: Fact Aware Neural Abstractive Summarization. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 32.

Yen-Chun Chen and Mohit Bansal. 2018. Fast Abstractive Summarization with Reinforce-Selected Sentence Rewriting. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 675–686, Melbourne, Australia. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net.

Dorottya Demszky, Kelvin Guu, and Percy Liang. 2018. Transforming Question Answering Datasets Into Natural Language Inference Datasets. ArXiv, abs/1809.02922.

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.

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

Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. 2021. GSum: A General Framework for Guided Neural Abstractive Summarization. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 4830–4842, 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.

Matan Eyal, Tat Baumel, and Michael Elhadad. 2019. Question Answering as an Automatic Evaluation Metric for News Article Summarization. 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 3938–3948, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Junxian He, Wojciech Kryściński, Bryan McCann, Nazneen Rajani, and Caiming Xiong. 2020. CTRLsum: Towards Generic Controllable Text Summarization.

Hanqi Jin, Tianming Wang, and Xiaojun Wan. 2020. SemSUM: Semantic Dependency Guided Neural Abstractive Summarization. In Proc. of the Conference on Artificial Intelligence (AAAI).

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7871–7880, Online. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In Text Summarization Branches Out, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In Proc. of the International Conference on Learning Representations.

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.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. SummaRuNNer: A Recurrent Neural Network Based Sequence Model for Extractive Summarization of Documents. In Proc. of the Conference on Artificial Intelligence (AAAI).

Ramesh Nallapati, Bowen Zhou, Cícero Nogueira dos Santos, Çağlar Gülçehre, 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, CoNLL 2016, Berlin, Germany, August 11-12, 2016, pages 280–290. ACL.
Feng Nan, Ramesh Nallapati, Zhiguo Wang, Cícero 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.

Shashi Narayan, Yao-Dong Zhao, Joshua Maynez, Gonçalo Simoes, Vitaly Nikolaev, and Ryan T. McDonald. 2021. Planning with Learned Entity Prompts for Abstractive Summarization.

Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. Data-to-Text Generation with Content Selection and Planning. In Proceedings of the AAAI conference on artificial intelligence, volume 33, pages 6908–6915.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know What You Don’t Know: Unanswerable Questions for SQuAD. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Alex Rush, er M., Sumit Chopra, and Jason Weston. 2015. A Neural Attention Model for Abstractive Sentence Summarization. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 379–389, Lisbon, Portugal. Association for Computational Linguistics.

Evan Sandhaus. 2008. The New York Times Annotated Corpus. Linguistic Data Consortium, Philadelphia, 6(12):e26752.

Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, Jacopo Staiano, Alex Wang, and Patrick Gallinari. 2021. QuestEval: Summarization Asks for Fact-based Evaluation. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 6594–6604, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Thomas Scialom, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. 2019. Answers Unite! Unsupervised Metrics for Reinforced Summarization Models. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 3246–3256, Hong Kong, China. Association for Computational Linguistics.

Ori Shapira, Hadar Ronen, Meni Adler, Yael Amstemdamer, Judit Bar-Ilan, and Ido Dagan. 2017. Interactive Abstractive Summarization for Event News Tweets. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 109–114, Copenhagen, Denmark. Association for Computational Linguistics.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and Answering Questions to Evaluate the Factual Consistency of Summaries. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 5008–5020, Online. Association for Computational Linguistics.

Johnny Wei and Robin Jia. 2021. The Statistical Advantage of Automatic NLG Metrics at the System Level. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 6840–6854, Online. Association for Computational Linguistics.

Daniela Brook Weiss, Paul Roit, Ayal Klein, Ori Ernst, and Ido Dagan. 2021. QA-Align: Representing Cross-Text Content Overlap by Aligning Question-Answer Propositions. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 9879–9894, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. In 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net.

Chenguang Zhu, William Hinthorn, Ruochen Xu, Qingkai Zeng, Michael Zeng, Xuedong Huang, and Meng Jiang. 2021. Enhancing Factual Consistency of Abstractive Summarization. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 718–733, Online. Association for Computational Linguistics.
Table 4: The number of instances in the training, validation, and test splits of the three datasets used in our experiments as well as the number of spans selected by the classification component that were passed as input to the generation component.

A Dataset Statistics

The sizes of the CNN/DailyMail, XSum, and NYTimes datasets are included in Table 4. The Table also includes the number of spans per span type that were selected from the classification component and passed to the generation component during inference. The values were selected based on a parameter sweep on the validation set. The number of spans with the highest ROUGE-1 F1 score was selected.

B Implementation Details

All of the models were trained with the same hyperparameters for across datasets and span types which were based on those used by BART (Lewis et al., 2020).

The classification component was a BART-Large model that was fine-tuned with a binary cross-entropy classification loss. We selected the model based on which had the best precision@1 on the validation dataset.

The generation models were also fine-tuned BART-Large models, but they instead use a cross-entropy loss function.

Both the components were trained using Adam (Loshchilov and Hutter, 2019) with weight decay and learning rate 3e-5. The classification component was trained for 3 epochs, and the final model was selected based on the precision@1 on the validation set. The generation component was trained for 5 epochs, and the final model was selected based on the ROUGE-2 F1 score on the validation set.

| Dataset  | #Train | #Valid | #Test | Span Type | #Spans |
|----------|--------|--------|-------|-----------|--------|
| CNN/DM   | 287,113| 13,368 | 11,490| Sentences | 3      |
|          |        |        |       | Entities  | 10     |
|          |        |        |       | NPs       | 25     |
|          |        |        |       | QA        | 20     |
| XSum    | 204,045| 11,332 | 11,334| Sentences | 1      |
|          |        |        |       | Entities  | 1      |
|          |        |        |       | NPs       | 5      |
|          |        |        |       | QA        | 1      |
| NYTimes | 44,382 | 5,523  | 6,495 | Sentences | 4      |
|          |        |        |       | Entities  | 15     |
|          |        |        |       | NPs       | 45     |
|          |        |        |       | QA        | 27     |

Table 5: The automatic evaluation metrics for summary quality are nearly the same for the QA-based model and the QA-based model trained on the augmented data.

C Data-Augmentation Automatic Evaluation

Table 5 contains the comparison between the standard and data-augmented training procedures based on the automatic metrics. The scores are nearly the same. The benefit of the model trained on the augmented data is in its controllability, which is not captured by this evaluation because the models trained with the standard and augmented training data receive the same spans as input supervision.

D Human Evaluation Details

Fig. 7 contains a screenshot of the tool we used for annotating summary quality on MTurk. The annotators were instructed to rate the summaries from “Very Poor” to “Very Good” based on whether the summary contained important information, was faithful to the input document, was fluent, and was cohesive. The ratings were converted to a Likert scale from 1-5 and averaged across all of the ratings for a system.

In order to encourage the annotators to pay attention to the task, we also required that they write a very brief explanation of how they made their decision, inspired by Narayan et al. (2021).
Figure 7: A screenshot of the tool we used for annotating summary quality on MTurk.