Incorporating Question Answering-Based Signals into Abstractive Summarization via Salient Span Selection

Daniel Deutsch[†]
Google
dandeutsch@google.com

Dan Roth
University of Pennsylvania
danroth@seas.upenn.edu

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]

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.

Figure 1: Salient spans identified by QA-based signals (shown in color) more precisely identify salient document content than those that identify salient sentences based on lexical overlap (shown in bold). Our method classifies the salient spans, marks them in the input document, and then generates a summary.

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

[1] http://cogcomp.org/page/publication_view/997
[†] Work done while at the University of Pennsylvania

Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, pages 575–588

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 generated wh-question is correctly answered in the gold summary according to a learned 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).

While we show that marking NPs as salient controls the summary content, the model often outputs extra, undesired information. To that extent, we propose a data augmentation procedure that removes sentences unsupported by any salient span and generates new training examples based on what content should be able to be generated by subsets of the salient spans (§4).

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 (§6.1). Further, our data augmentation procedure results in summaries that are significantly shorter with only a small reduction in the percent of target content covered, demonstrating it successfully eliminates undesired summary content (§6.2).

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 QA is used 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

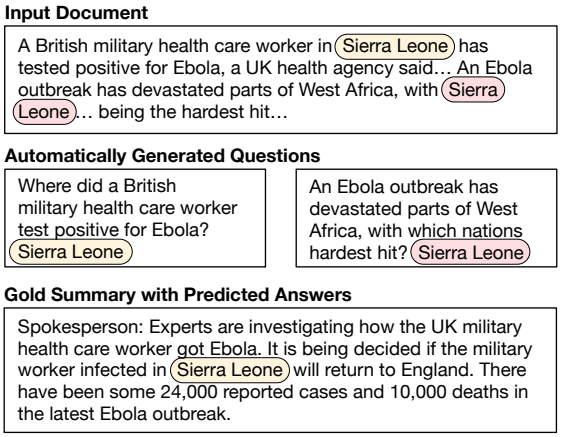

Figure 2: We define a document noun phrase as salient if the wh-question it answers is also answered in the gold summary. Here, the first (yellow) instance of "Sierra Leone" is salient and the second (red) is not because the gold summary answers the automatically generated question for the first instance but not the second.

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 QA 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.

We refer to the NPs identified by this procedure as "silver spans."[2] Specific implementation details

[2]The term "silver" refers to the fact that the salient spans are not perfect because they were identified by sequence of learned models rather than humans ("gold" spans; §6.1).

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.

3 A Two-Stage, Span-Based Model

Next, we propose a two-stage, span-based model 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 silver 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 QG-QA 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 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 silver spans and the ground-truth summary to fine-tune BART using a standard cross-entropy loss function. The ground-truth summary does not have any marking of salient spans. For inference, the predicted salient spans from the span classifier are used instead of the silver spans.

4 Improving Controllability via Data Augmentation

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. Indeed, we later show in §6.2 that this is true, thus the content of the summary can be controlled by which spans are marked as salient. However, it is also desirable for a controllable summarization model to also not include

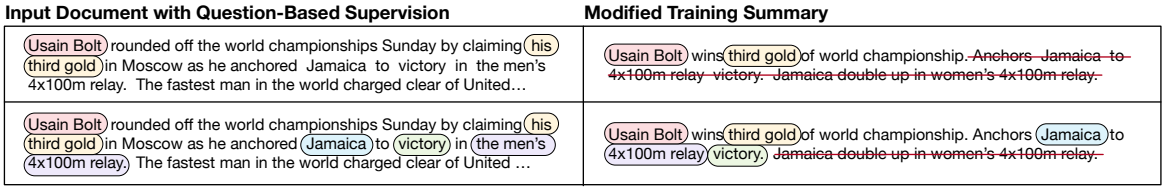

Figure 3: An example of our data augmentation procedure. The colors represent the mapping between document and summary spans. The document spans are given to the generation model during training. In this example, no span maps to the third summary sentence, so it is removed entirely. Then, new training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

content which is not marked as salient. The generation models 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 silver salient spans that were used to train the generation model, so it may learn to output extra, unmarked information. This could happen if the QG/QA models are imperfect (resulting in a noisy supervision signal) 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 silver span annotation procedure enables us to address these controllability issues. 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 QG-QA 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 has no phrase mapped to the document. These sentences would encourage the model to generate additional content based on unmarked spans.

Second, we generate new pairs of salient spans and gold summaries for training by selecting the first k remaining gold summary sentences and the subset of salient document spans which map to them. For instance, if $k = 2$, only the salient spans which are mapped to the first two summary sentences are marked in the input document, and the model is trained to generate only those sentences. We generate new examples for each original training instance using all possible values of k. By training on these new pairs, the model should learn to better control the length of the output summary based on the number of marked salient spans. An example of these augmentations is included in Fig. 3.

Although this procedure is described within the context of the QA-based supervision, it can be implemented with any such mapping between the document and gold summaries.

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 it. 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 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 QA-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 G.

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 QA 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 QG/QA 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 Summarization Evaluation

Automatic Evaluation Table 1 contains the models' performances as evaluated by automatic metrics, both using the spans predicted by the classifier ("end-to-end") and the silver spans (i.e., assuming a "perfect" classifier).

Interestingly, we find a somewhat surprising result. On CNN/DailyMail and NYTimes, the end-to-end QA-based model performs the best among the different span labeling methods and the baseline BART. On NYTimes, it is also better than GSum. However, if the silver span labels are used, the lexical NP-based model out-performs the rest by a somewhat large margin. It is surprising that a seemingly better generation model would result in worse end-to-end performance.

To better understand this result, we manually labeled all of the NPs in 50 CNN/DailyMail documents as salient or not salient based on whether the corresponding predicate-argument relation was present in the reference summary (see Appendix E for details). These spans, which we call the gold spans, can be used to evaluate the precision and recall of the silver spans as well as the output from the salient span classifiers.

Table 2 shows that the QA-based labels are more precise but have lower recall than the lexical NP labels. Because the lexical NP method aggressively marks the first occurrence of any NP in the document which is present in the reference as salient, it is unsurprising that its recall would be high. Since it cannot distinguish between instances of the same NP due to its bag-of-words representation, its precision is low. In contrast, the QA-based approach can reason about which occurrence of an NP is salient (resulting in higher precision), but the recall is lower likely due to noise in the QG/QA models. This same pattern appears in Table 2 with the outputs from the salient span classifiers, although the precisions and recalls are notably lower than the silver span labels'.

We believe that the discrepancy between the end-to-end and silver span-based models' performances can be explained by these results. The lexical NP generation model was trained with a high recall silver supervision at 66.3, allowing the generation

Method	CNN/DailyMail					NYTimes				
	R1	R2	RL	BSc	QAE	R1	R2	RL	BSc	QAE
Baselines & Other Work										
BART	44.2	21.3	40.9	-	-	-	-	-	-	-
BART (ours)	44.1	21.0	40.9	88.3	23.5	54.0	35.2	50.7	89.5	27.3
GSum	46.0†	22.3†	42.6†	88.6†	22.9	54.3	35.4	47.6	-	-
Silver Spans										
Sentences	51.7	29.9	48.8	89.4	28.6	62.7	46.0	59.8	91.2	33.5
Entities	51.5	27.6	48.0	89.6	30.0	60.9	42.8	57.6	90.8	32.0
Lexical NPs	**59.6**	**34.6**	**55.8**	**90.6**	**36.2**	**68.2**	**50.7**	**64.8**	**92.0**	**36.6**
QAs	55.3	31.4	51.9	90.0	33.7	65.7	48.7	62.6	91.6	35.8
End-to-End										
Sentences	45.0	**21.8**	41.8	88.2	23.2	54.6	35.9	51.4	89.6	27.6
Entities	43.5	20.3	40.4	88.3	23.2	53.5	34.6	50.3	89.4	27.0
Lexical NPs	44.8	21.0	41.6	88.4	23.2	54.6	35.4	51.3	89.6	27.1
QAs	**45.5**	**21.9**	**42.4**†	**88.5**	**24.4**†	**55.2**†	**36.3**†	**51.9**†	**89.7**†	**28.0**†

Table 1: The automatic metric results for the baselines and other work (top), models that use silver 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 silver labels) using a permutation test with $\alpha = 0.05$.

Method	Precision	Recall	F_1
Silver Labels			
Lexical NPs	32.7	**66.3**	41.8
QAs	**43.8**	51.5	**45.3**
Predicted Spans			
Lexical NPs@25	23.8	**54.4**	**32.0**
QAs@20	**27.3**	49.1	**33.8**

Table 2: The average summary-level precision, recall, and F_1 scores of the silver labeling methods (top) and the output from the span classifiers (bottom) evaluated against the human-annotated gold labeling. Results in bold are statistically higher (or tied) under a single-tail pairwise permutation test with $\alpha = 0.05$. The @k values were selected based on validation set performance.

Method	CNN/DailyMail		
	R1	R2	RL
Silver Spans			
Lexical NPs	**59.6**	**34.6**	**55.8**
+10% Noise	57.8	32.8	54.0
+20% Noise	56.3	31.5	52.6
+30% Noise	55.0	30.4	51.4
+35% Noise	54.1	29.6	50.6
QAs	55.3	31.4	51.9
End-to-End			
Lexical NPs	44.8	21.0	41.6
+10% Noise	45.0	21.3	41.8
+20% Noise	45.2	21.6	42.0
+30% Noise	45.3	21.7	42.1
+35% Noise	45.1	21.6	41.9
QAs	**45.5**	**21.9**	**42.4**

Table 3: The ablated lexical NP supervision shows as the noise increases, the silver span performance decreases but end-to-end performance improves.

model to achieve good performance when the silver spans are provided. Yet during inference the model is provided with spans that only have around 54.4 recall, 12 points lower. We suspect the generation model learned to rely heavily on the marked salient spans — and empirically we observed that it copied very heavily from them — thus when the quality of the span signal was reduced, the resulting summaries similarly got worse. In contrast, the difference between the QA-based model's recall during training and inference is only estimated to be around 2.4, so this issue is less severe, resulting in better end-to-end summaries.

To test this hypothesis, we artificially ablated the lexical NP-based generation model's silver span supervision's recall by removing $k\%$ of the salient spans uniformly at random — thus making the training spans look more similar to the spans during inference — and retrained the model. We would expect the silver span-based model's performance to decrease while the end-to-end model's increases. Indeed, in Table 3 we find that this does happen. These results suggest that the relationship between the classifier's performance and generation model's supervision is important for good end-to-end results and could be explored in future work.

Although the end-to-end lexical NP results begin

	BART	Sentences	QA
Quality Score	3.76	**3.86**	**4.00**

Table 4: Summary quality scores according to humans. Results in bold are statistically tied for the best score.

to approach the QA-based model's performance, they do not quite reach it. Further, the QA-based silver spans maintain an F_1 advantage over the lexical NP method (Table 2). While the QA-based approach can be improved with better question generation and answering models, the lexical NP labeling method is inherently limited. Therefore, the QA-based method does appear to be the best method of incorporating additional supervision into the summarization models based on the automatic metrics.

Human Evaluation Table 4 contains the results of evaluating BART and the sentence- and QA-based models on CNN/DailyMail (the best performing) 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).

6.2 Controllability Evaluation

Automatic Evaluation The controllability of the QA-based generation model is evaluated in Fig. 4 using the original training data as well as the augmented data described in §4. 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 salient span classifier for various values of k on CNN/DailyMail. The data augmentation procedure is split into 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.

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.

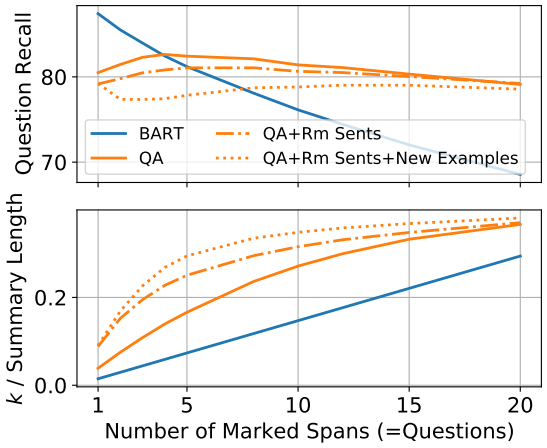

Figure 4: The percent of questions which correspond to the marked spans answered by the generated summaries (top) and the summary lengths in tokens (bottom). The QA methods have higher question recall than BART and are far more concise, demonstrating that marking input spans controls the summary content.

Among the QA-based models, we do observe a 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. 5. Because the sentence-based model is limited to marking full sentences, the content which is taken from the marked sentence cannot be further

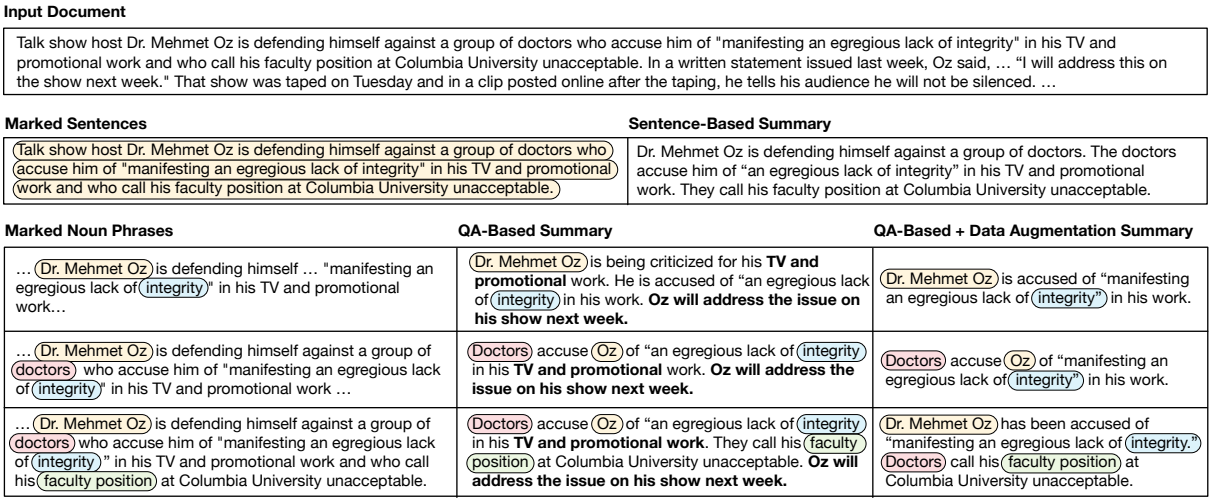

Figure 5: Example summaries generated by the sentence-based model (middle), QA-based model (bottom center) and QA-based model trained on the augmented data (bottom right). The QA-based models allow for much more control over the summary content than the sentence model by marking different combinations of phrases. The augmented-data summaries better eliminate unmarked content from the input than the standard model (extra information generated by the standard model shown in bold).

controlled. In contrast, the figure shows how the QA models' summaries can be altered by marking different NPs within the sentence, thus demonstrating the benefits of phrase-level controllability.

The example in Fig. 5 also shows how the data augmentation procedure 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.

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 document, 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 to generate 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. These works include that information to try and generate faithful summaries, whereas our goal is to identify salient document content.

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), marking records in a data-to-text approach (Puduppully et al., 2019), or using aspect controllers (Amplayo

et al., 2021). 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 summaries than 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.

Acknowledgments

The authors would like to thank the anonymous EACL reviewers for their insightful feedback on our work.

This work was supported by a Focused Award from Google and Contracts FA8750-19-2-1004 and FA8750-19-2-0201 with the US Defense Advanced Research Projects Agency (DARPA). Approved for Public Release, Distribution Unlimited. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

Limitations

The summarization model proposed in this work relies on the existence of question generation and question answering models, which are used to preprocess the data for training. It is likely that such models only exist or are of high-enough quality for high resource languages, such as English, which limits the languages that our model can be trained on in practice. However, we do not see a reason why our model could not be applied to another language as long as those additional resources are available.

Further, because the question generation model produces simple wh-questions and the question answering model is only able to reason about the predicate-argument structure of the text (due to it being trained on SQuAD 2.0), our procedure for identifying salient document phrases requires that the same information across the document and reference summary must be expressed in relatively similar ways (e.g., up to rephrasing and synonyms). If the reference summary does contain the answer to the generated question but identifying that answer requires a level of reasoning beyond reasoning about the predicate-argument structure (e.g., does it require multi-hop reasoning?), the specific models proposed in this work may fail to identify those salient phrases. This limits the types of datasets for which we expect our proposal to do well on (discussed more in Appendix D). However, if matching document and reference summary information requires a level of reasoning that is supported by the generation and answering models, then we suspect our proposal will work, in theory, but we have not experimented with this in practice.

Our methods out-perform the baseline systems the most when evaluated by QAEval, and we leverage the same QG/QA technology as this metric. While this commonality may bias our system toward summaries that are favored by QAEval, we argue this is not necessarily a bad thing. Previous work has incorporated ROUGE-based signals into their models, either indirectly by selecting extractive labels based on ROUGE or directly by optimizing ROUGE via reinforcement learning. Our approach is analogous to these modeling approaches, and QAEval has been demonstrated to be a better evaluation metric than ROUGE, so it is likely a better metric to optimize for.

References

Reinald Kim Amplayo, Stefanos Angelidis, and Mirella Lapata. 2021. Aspect-Controllable Opinion Summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6578–6593, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Kristjan Arumae and Fei Liu. 2018. Reinforced Extractive Summarization with Question-Focused Re-

wards. 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 Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2566–2577, Minneapolis, Minnesota. Association for Computational Linguistics.

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, Tal 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, Cicero dos Santos, Çağlar Gulçehre, 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, Berlin, Germany. Association for Computational Linguistics.

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

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Shashi Narayan, Yao-Dong Zhao, Joshua Maynez, Gonccalo 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.

Alexander M. Rush, 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 Amsterdamer, 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.

A Dataset Statistics

The sizes of the CNN/DailyMail, XSum, and NYTimes datasets are included in Table 5. The Table

Dataset	#Train	#Valid	#Test	Span Type	#Spans
CNN/DM	287,113	13,368	11,490	Sentences	3
				Entities	10
				Lex. 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
				Lex. NPs	45
				QA	27

Table 5: 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.

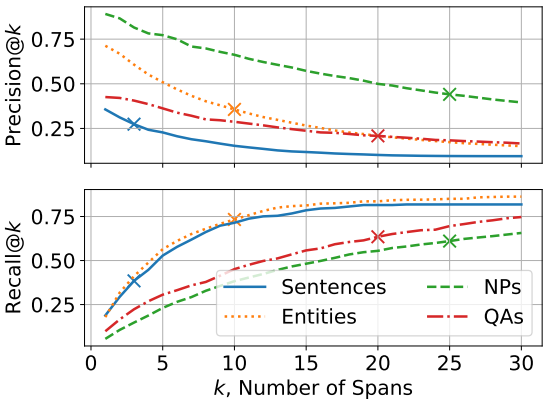

Figure 6: The performances of the salient span classifiers using the different types of salient phrase labeling evaluated against the silver spans. The "x" marks the operating points used in the end-to-end models.

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-2 F_1 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 F_1 score on the validation set.

C Salient Span Classifier Evaluation

Fig. 6 contains the precision@k and recall@k of the span based classifiers calculated against the corresponding silver spans. These plots should be interpreted as how well the span classifiers were able to learn from their respective supervision, not necessarily the true quality of the output span labels (which would require evaluating against human-annotated gold labels, as in Table 2). 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-2 F_1 score on the validation data.

D XSum Results

Table 6 contains the automatic summarization evaluation results on the XSum dataset. These results are included in the Appendix because incorporating the span-based supervision does not improve end-to-end results over the baseline BART model, which is a conclusion also reached by GSum, a model closely related to ours.

We suspect this is due to the abstractive nature of XSum compared to the more extractive CNN/DailyMail and NYTimes. Since the methods for identifying salient spans rely on the document and gold summary explicitly stating the salient content, we suspect the abstractiveness of XSum would result in this happening less frequently and thus be less beneficial to a summarization model trained on XSum.

E Gold Span Annotation Protocol

We selected 50 test instances from the CNN/DailyMail dataset uniformly at random and labeled each of the document NPs as salient or

Method	XSum				
	R1	R2	RL	BSc	QAE
Baselines & Other Work					
BART	45.1	21.3	40.9	-	-
BART (ours)	45.7†	22.4†	37.2†	91.3†	18.9†
GSum	44.9	21.2	36.0	90.4	17.9
Silver Spans					
Sentences	47.3	24.2	38.7	91.5	19.9
Entities	48.1	24.2	39.1	91.7	21.3
Lexical NPs	**54.3**	**29.3**	**44.1**	**92.4**	**26.1**
QAs	47.9	24.1	39.2	91.6	21.4
End-to-End					
Sentences	**45.0**	**21.7**	**36.6**	**91.2**	**18.6**
Entities	44.1	20.9	35.9	91.0	17.6
Lexical NPs	42.5	19.2	34.2	90.8	16.4
QAs	**45.1**	**21.8**	**36.7**	91.2	17.9

Table 6: The results of the models trained on the XSum dataset as evaluated with the automatic evaluation metrics. The span-based models do not improve over the baseline BART, potentially due to the abstractive nature of the XSum dataset.

not salient based on whether the corresponding predicate-argument relation also appears in the gold summary. We did not mark instances in which the NP's predicate-argument relation could be inferred from the gold summary via entailment as salient since our silver span labeling methods aim to mark phrases as salient if the content is explicilty included in the gold summary.

In general, this procedure was straightforward due to the extractive nature of the dataset in which the gold summaries copy heavily from the input document. If information was repeated in the input document, we tried to label the occurrence which contained the most predicate-argument relations which also matched the gold summary. That is, we selected the "best match." Otherwise, the first occurrence was selected.

Although our labeling procedure may be noisy, we do not have reason to believe that the labels may be biased in favor of either the lexical NP or QA labeling methods. Therefore, the statistics calculated from these labels should only be used as diagnostic tools to make relative comparisons between the different labeling methods rather than precise estimates of their exact values. 50 documents were sufficient to achieve statistically different results.

Our annotations will be released after publication.

Model	R1	R2	RL	BSc	QAE
Silver Spans					
QAs	55.3	31.4	51.9	90.0	33.7
QAs + Data Aug.	55.2	31.3	51.7	89.9	33.4
End-to-End					
QAs	45.5	21.9	42.4	88.5	24.4
QAs + Data Aug.	45.3	21.8	42.1	88.4	24.3

Table 7: 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.

F Data-Augmentation Automatic Evaluation

Table 7 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.

G 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).

The MTurk annotators were paid at a rate of around $15 USD per hour.

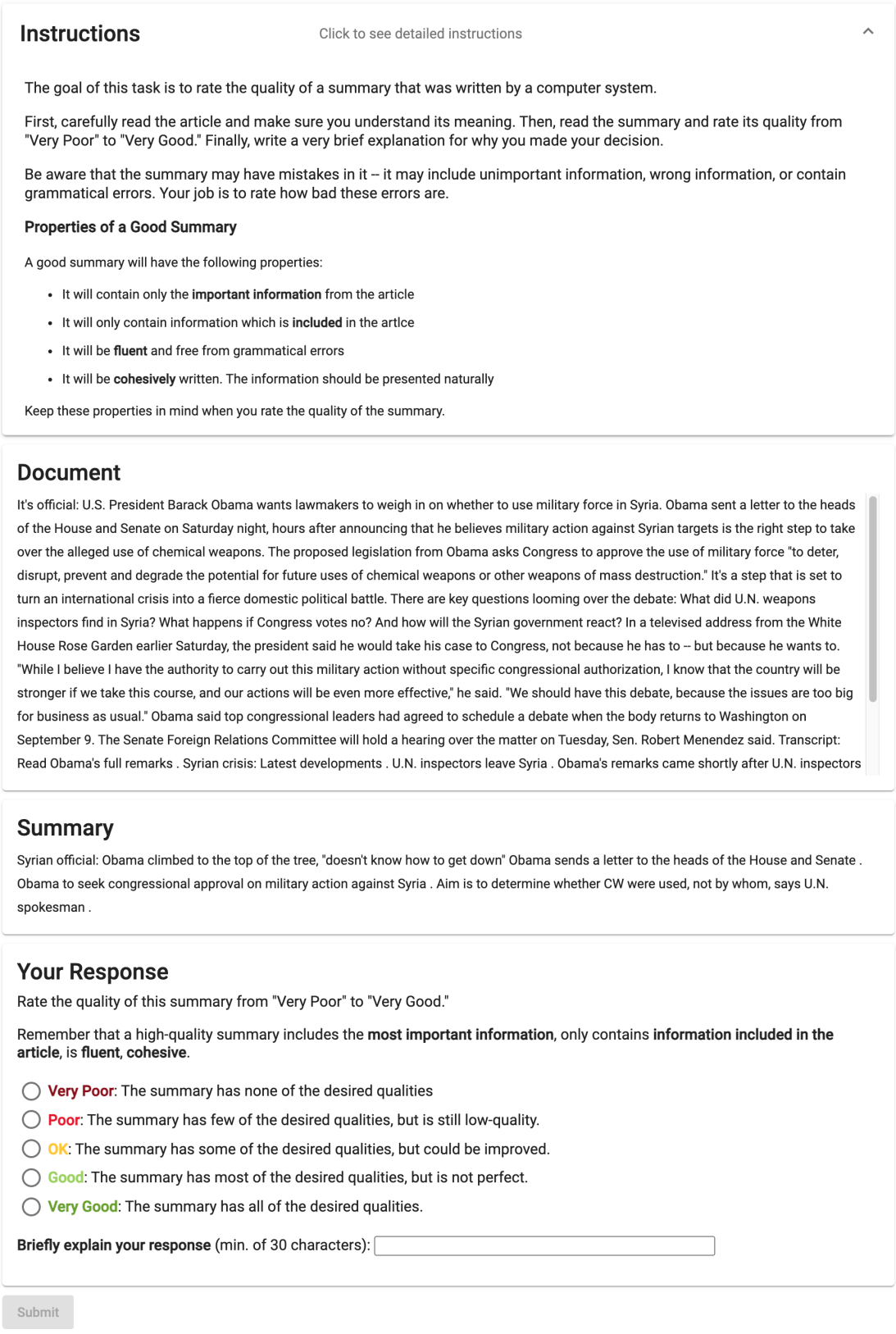

Figure 7: A screenshot of the tool we used for annotating summary quality on MTurk.