# Multi-Fact Correction in Abstractive Text Summarization

**Yue Dong**[1]* **Shuohang Wang**[2] **Zhe Gan**[2] **Yu Cheng**[2]
**Jackie Chi Kit Cheung**[1] **Jingjing Liu**[2]

[1]Mila / McGill University, [2]Microsoft Dynamics 365 AI Research
{yue.dong2@mail, jcheung@cs}.mcgill.ca
{shuowa, zhe.gan, yu.cheng, jingjl }@microsoft.com

## Abstract

Pre-trained neural abstractive summarization systems have dominated extractive strategies on news summarization performance, at least in terms of ROUGE. However, system-generated abstractive summaries often face the pitfall of factual inconsistency: generating incorrect facts with respect to the source text. To address this challenge, we propose *SpanFact*, a suite of two factual correction models that leverages knowledge learned from question answering models to make corrections in system-generated summaries via *span selection*. Our models employ single or multi-masking strategies to either iteratively or autoregressively replace entities in order to ensure semantic consistency w.r.t. the source text, while retaining the syntactic structure of summaries generated by abstractive summarization models. Experiments show that our models significantly boost the factual consistency of system-generated summaries without sacrificing summary quality in terms of both automatic metrics and human evaluation.

## 1 Introduction

Informative text summarization aims to shorten a long piece of text while preserving its main message. Existing systems can be divided into two main types: extractive and abstractive. Extractive strategies directly copy text snippets from the source to form summaries, while abstractive strategies generate summaries containing novel sentences not found in the source. Despite the fact that extractive strategies are simpler and less expensive, and can generate summaries that are more grammatically and semantically correct, abstractive strategies are becoming increasingly popular thanks to its flexibility, coherency and vocabulary diversity (Zhang et al., 2020a).

*Most of this work was done when the first author was an intern at Microsoft.

| | |
|---|---|
| CNNDM Source | (CNN) About a quarter of a million Australian homes and businesses have no power after a "once in a decade" storm battered Sydney and nearby areas. About 4,500 people have been isolated by flood waters as "the roads are cut off and we won't be able to reach them for a few days,"... |
| Bottom-up Summary | a quarter of a million australian homes and businesses have no power after a decade. |
| Corrected by SpanFact | about a quarter of a million australian homes and businesses have no power after a "once in a decade" storm. |
| Gigaword Source | all the 12 victims including 8 killed and 4 injured have been identified as senior high school students of the second senior high school of ruzhou city, central china's henan province, local police said friday. |
| Pointer-Generator Summary | 12 killed, 4 injured in central china school shooting. |
| Corrected by SpanFact | 8 killed, 4 injured in central china school shooting. |
| XSum Source | st clare's catholic primary school in birmingham has met with equality leaders at the city council to discuss a complaint from the pupil's family. the council is supporting the school to ensure its policies are appropriate... |
| BertAbs Summary | a muslim school has been accused of breaching the equality act by refusing to wear headscarves. |
| Corrected by SpanFact | a catholic school has been accused of breaching the equality act by refusing to wear headscarves. |

Table 1: Examples of factual error correction on different summarization datasets. Factual errors are marked in red. Corrections made by the proposed SpanFact models are marked in orange.

Recently, with the advent of Transformer-based models (Vaswani et al., 2017) pre-trained using self-supervised objectives on large text corpora (Devlin et al., 2019; Radford et al., 2018; Lewis et al., 2020; Raffel et al., 2020), abstractive summarization models are surpassing extractive ones on automatic evaluation metrics such as ROUGE (Lin, 2004). However, several studies (Falke



*Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 9320–9331,
November 16–20, 2020. 

et al., 2019; Goodrich et al., 2019; Kryściński et al., 2019; Wang et al., 2020; Durmus et al., 2020; Maynez et al., 2020) observe that despite high ROUGE scores, system-generated abstractive summaries are often factually inconsistent with respect to the source text. Factual inconsistency is a well-known problem for conditional text generation, which requires models to generate readable text that is faithful to the input document. Consequently, sequence-to-sequence generation models need to learn to balance signals between the source for faithfulness and the learned language modeling prior for fluency (Kryściński et al., 2019). The dual objectives render abstractive summarization models highly prone to hallucinating content that is factually inconsistent with the source documents (Maynez et al., 2020).

Prior work has pushed the frontier of guaranteeing factual consistency in abstractive summarization systems. Most focus on proposing evaluation metrics that are specific to factual consistency, as multiple human evaluations have shown that ROUGE or BERTScore (Zhang et al., 2020b) correlates poorly with faithfulness (Kryściński et al., 2019; Maynez et al., 2020). These evaluation models range from using fact triples (Goodrich et al., 2019), textual entailment predictions (Falke et al., 2019), adversarially pre-trained classifiers (Kryściński et al., 2019), to question answering (QA) systems (Wang et al., 2020; Durmus et al., 2020). It is worth noting that QA-based evaluation metrics show surprisingly high correlations with human judgment on factuality (Wang et al., 2020), indicating that QA models are robust in capturing facts that can benefit summarization tasks.

On the other hand, some work focuses on model design to incorporate factual triples (Cao et al., 2018; Zhu et al., 2020) or textual entailment (Li et al., 2018; Falke et al., 2019) to boost factual consistency in generated summaries. Such models are efficient in boosting factual scores, but often at the expense of significantly lowering ROUGE scores of the generated summaries. This happens because the models struggle between generating pivotal content while retaining true facts, often with an eventual propensity to sacrificing informativeness for the sake of correctness of the summary. In addition, these models inherit the backbone of generative models that suffer from hallucination despite the regularization from complex knowledge graphs or text entailment signals.

In this work, we propose SpanFact, a suite of two neural-based factual correctors that improve summary factual correctness without sacrificing informativeness. To ensure the retention of semantic meaning in the original documents while keeping the syntactic structures generated by advanced summarization models, we focus on factual edits on entities only, a major source of hallucinated errors in abstractive summarization systems in practice (Kryściński et al., 2019; Maynez et al., 2020). The proposed model is inspired by the observation that fact-checking QA model is a reliable medium in assessing whether an entity should be included in a summary as a fact (Wang et al., 2020; Durmus et al., 2020). To our knowledge, we are the first to adapt QA knowledge to enhance abstractive summarization. Compared to sequential generation models that incorporate complex knowledge graph and NLI mechanisms to boost factuality, our approach is lightweight and can be readily applied to any system-generated summaries without retraining the model. Empirical results on multiple summarization datasets show that the proposed approach significantly improves summarization quality over multiple factuality measures without sacrificing ROUGE scores.

Our contributions are summarized as follows. ($i$) We propose SpanFact, a new factual correction framework that focuses on correcting erroneous facts in generated summaries, generalizable to any summarization system. ($ii$) We propose two methods to solve multi-fact correction problem with single or multi-span selection in an iterative or auto-regressive manner, respectively. ($iii$) Experimental results on multiple summarization benchmarks demonstrate that our approach can significantly improve multiple factuality measurements without a huge drop on ROUGE scores.

## 2 Related Work

The general neural-based encoder-decoder structure for abstractive summarization is first proposed by Rush et al. (2015). Later work improves this structure with better encoders, such as LSTMs (Chopra et al., 2016) and GRUs (Nallapati et al., 2016), that are able to capture long-range dependencies, as well as with reinforcement learning methods that directly optimize summarization evaluation scores (Paulus et al., 2018). One drawback of the earlier neural-based summarization models is the inability to produce out-of-

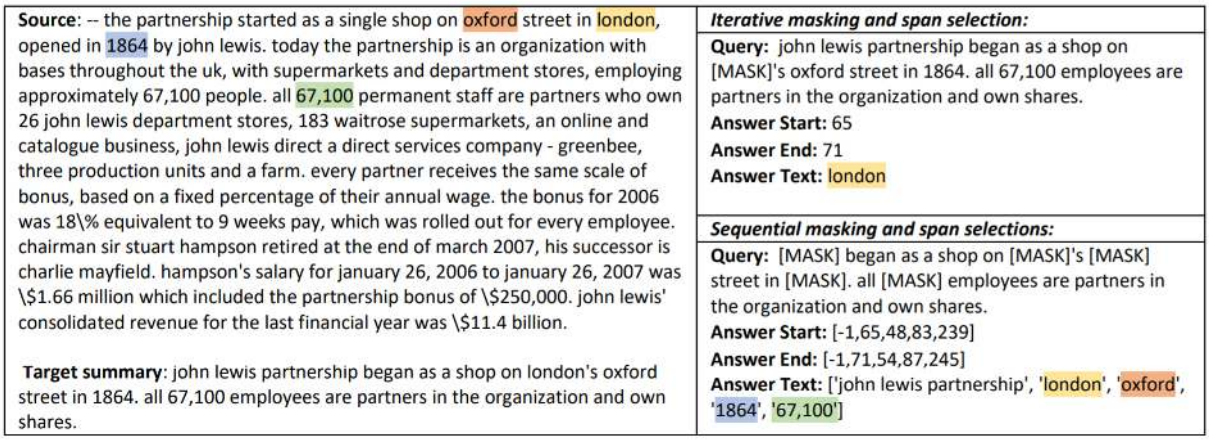


Figure 1: Training example created for the QA-span prediction model (upper right) and the auto-regressive fact correction model (bottom right).

vocabulary words, as the model can only generate whole words based on a fixed vocabulary. See et al. (2017) proposes a pointer-generator framework that can copy words directly from the source through a pointer network (Vinyals et al., 2015), in addition to the traditional sequence-to-sequence generation model.

Abstractive summarization starts to shine with the advent of self-supervised algorithms, which allow deeper and more complicated neural networks such as Transformers (Vaswani et al., 2017) to learn diverse language priors from large-scale corpora. Models such as BERT (Devlin et al., 2019), GPT (Radford et al., 2018) and BART (Lewis et al., 2020) have achieved new state-of-the-art performances on abstractive summarization (Liu and Lapata, 2019; Lewis et al., 2020; Zhang et al., 2020a; Shi et al., 2019; Fabbri et al., 2019). These models often finetune pre-trained Transformers with supervised summarization datasets that contain pairs of source and summary.

However, encoder-decoder architectures widely used in abstractive summarization systems are inherently difficult to control and prone to hallucination (Vinyals and Le, 2015; Koehn and Knowles, 2017; Lee et al., 2018), and often leads to factual inconsistency: the system-generated summary is fluent but unfaithful to the source (Cao et al., 2018). Studies have shown that 8% to 30% system-generated abstractive summaries have factual errors (Falke et al., 2019; Kryściński et al., 2019) that cannot be discovered by ROUGE scores. Recent studies have proposed new methods to ensure factual consistency in summarization. Cao et al. (2018); Zhu et al. (2020) propose RNN-based and Transformer-based decoders that attend to both source and extracted knowledge triples, respectively. Li et al. (2018) propose an entailment-reward augmented maximum-likelihood training objective, and Falke et al. (2019) proposes to rerank beam results based on entailment scores to the source.

Our fact correction models are inherently different from these models, as we focus on post-correcting summaries generated by any model. Our models are trained with the objective of predicting masked entities identified for fact correction (Figure 1), and learn to fill in the entity masks of any system-generated summaries with single or multi-span selection mechanism (Figure 2). The most similar work to ours is proposed concurrently by Meng et al. (2020), where they fine-tune a BART (Lewis et al., 2020) model on distant supervision examples and use it as a post-editing model for factual error correction.

## 3 Multi-Fact Correction Models

In this section, we describe two models proposed for factual error correction: $(i)$ QA-span Fact Correction model, and $(ii)$ Auto-regressive Fact Correction model. As both methods rely on span selection with different masking and prediction strategies, we call them SpanFact collectively.

### 3.1 Problem Formulation

Let $(x, y)$ be a document-summary pair, where $x = (x_1, \ldots, x_M)$ is the source sequence with $M$ tokens, and $y = (y_1, \ldots, y_N)$ is the target sequence with $N$ tokens. An abstractive summarization model aims to model the conditional

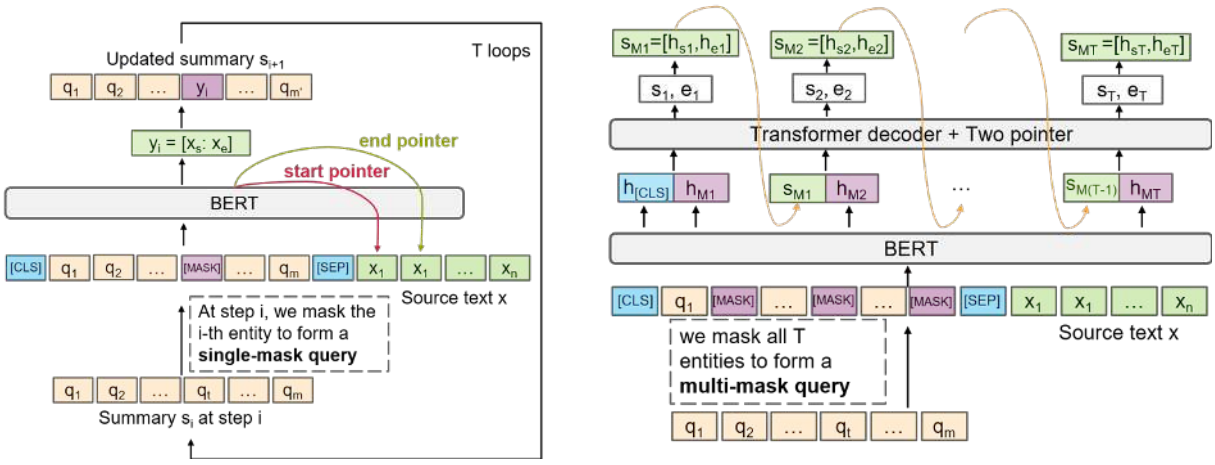


Figure 2: Model architecture (Left: QA-span fact correction model. Right: Auto-regressive fact correction model).

likelihood $p(y|x)$, which can be factorized into a product $p(y|x) = \prod_{t=1}^{T} p(y_t|y_{1....,t-1}, x)$, where $y_{1....,t-1}$ denote the preceding tokens before position $t$. The conditional maximum-likelihood objective ideally requires summarization models to not only optimize for informativeness but also correctness. However, in reality this often fails as the models have a high propensity for leaning towards informativeness than correctness (Li et al., 2018).

Suppose a summarization system generates a sequence of tokens $y' = (y'_1, \ldots, y'_N)$ to form a summary. Our factual correction models aim to edit an informative-yet-incorrect summary into $y'' = (y''_1, \ldots, y''_K)$ such that

$$f(x, y'') > f(x, y') \,, \tag{1}$$

where $f$ is a metric measuring factual consistency between the source and system summary.

### 3.2 Span Selection Dataset

Our fact correction models are inspired by the *span selection* task, which is often used in reading comprehension tasks such as question answering. Figure 1 shows examples of the *span selection* datasets we created for training our QA-span and auto-regressive fact correction models, respectively. The *query* is a reference summary masked with one or all entities,[1] and the passage is the corresponding source document to be summarized. If an entity appears multiple times in the source document, we rank them based on the fuzzy string-matching scores (a variation of Levenshtein distance) between the query sentence and the source sentence containing the entity. Our models explicitly learn to predict the span of the masked entity rather than pointing to a specific token as in Pointer Network (Vinyals et al., 2015), because the original tokens and replaced tokens often have different lengths.

Our QA-span fact correction model iteratively mask and replace one entity at a time, while the auto-regressive model masks all the entities simultaneously, and replace them in an auto-regressive fashion from left to right. Figure 2 shows an overview of our models. Comparing the two models, the QA-span fact correction model works better when only a few errors exist in the draft summary, as the prediction of each mask is relatively independent of each other. On the other hand, the auto-regressive fact correction model starts with a skeleton summary that has all the entities masked, which is often more robust when summaries contain many factual errors.

### 3.3 QA-Span Fact Correction Model

In the iterative setting, our model aims to conduct entity correction by answering a query that contains only one mask at a time. Suppose a system summary has $T$ entities. At time step $i$, we mask the $i$-th entity and use this masked sequence as the query to our QA-span model. The prediction is placed into the masked slot in the query to generate an updated system summary to be used in the next step.

Given the source text $x$ and a masked query $q = (y'_1, \ldots, \texttt{[MASK]}, \ldots y'_m)$, our iterative correction model aims to predict the answer span via modeling $p(i = \text{start})$ and $p(i = \text{end})$. For span

[1] In this work, we use SpaCy NER tagger (Honnibal and Montani, 2017) to identify entities for data construction.

selection, we use the BertForQuestionAnswering[2] model, which adds two separate non-linear layers on top of Transformers as pointers to the start and end token position for the answer. We initialize the fact-correction model from a pre-trained BERT model (Devlin et al., 2019), and perform finetuning with the span selection datasets we created from the summarization datasets (Figure 1).

The input to the BERT model is a concatenation of two segments: the masked query $q$ and the source $x$, separated by special delimiter markers as ($\texttt{[CLS]}, q, \texttt{[SEP]}, x$). Each token in the sequence is assigned with three embeddings: token embedding, position embedding, and segmentation embedding.[3] These embeddings are summed into a single vector and fed to the multi-layer Transformer model:

$$\tilde{\boldsymbol{h}}^l = \text{LN}(\boldsymbol{h}^{l-1} + \text{MHAtt}(\boldsymbol{h}^{l-1}))\,, \quad (2)$$

$$\boldsymbol{h}^l = \text{LN}(\tilde{\boldsymbol{h}}^l + \text{FFN}(\tilde{\boldsymbol{h}}^l))\,, \quad (3)$$

where $\boldsymbol{h}^0$ are the input vectors, and $l$ represents the depth of stacked layers. LN and MHAtt are layer normalization and multi-head attention operations (Vaswani et al., 2017). The top layer provides the hidden states for the input tokens with rich contextual information. The start (s) and end (e) of the answer span are predicted as:

$$a_i^{start} = p(i = s) = \frac{\exp(q_i^s)}{\sum_{j=0}^{H-1} \exp(q_j^s)}\,, \quad (4)$$

$$a_i^{end} = p(i = e) = \frac{\exp(q_i^e)}{\sum_{j=0}^{H-1} \exp(q_j^e)}\,, \quad (5)$$

$$q_i^s = \text{ReLU}(\boldsymbol{w}_s^\top \boldsymbol{h}_i + b_s)\,, \quad (6)$$

$$q_i^e = \text{ReLU}(\boldsymbol{w}_e^\top \boldsymbol{h}_i + b_e)\,, \quad (7)$$

where $H$ is the number of encoder's hidden states, $\boldsymbol{w}_s, \boldsymbol{w}_e \in \mathbb{R}^d$ and $b_s, b_e \in \mathbb{R}$ are trainable parameters. The final span is selected based on the argmax of Eqn. (4) and (5) with the constraint of $p_{start} < p_{end}$ and $p_{end} - p_{start} < k$.

### 3.4 Auto-regressive Fact Correction Model

One disadvantage of the QA-style span-prediction strategy is that if the sequence contains too many factual errors, masking out one entity at a time may lead to highly erroneous skeleton summary to start with. The model might be making predictions on top of wrong entities from later in the sequence. Masking one entity at a time is essentially a greedy local method that is prone to error accumulation. To alleviate this issue, we propose a new sequential fact correction model to handle errors in a more global manner with beam search. Specifically, we mask out all the entities simultaneously, and use a novel *auto-regressive* span-selection decoder to predict fillers for the multiple masks sequentially. By doing this, we assume dependency between the masks: the earlier predicted entities will be used as corrected context for better predictions in the later steps.

Given a source text $x = (x_1, \ldots, x_n)$ and a draft summary $(y'_1, \ldots y'_m)$. Our model first masks out all the entities (with $T$ masks), and leaves a skeleton summary as the query $q = (y'_1, \ldots, \texttt{[MASK]}_1, \ldots, \texttt{[MASK]}_T \ldots y'_m)$. Then, we concatenate the query $q$ with the source $x$ (similar to Section 3.3) as inputs to the encoder. The inputs are fed into BERT to obtain contextual hidden representations.

We then *select* the encoder's hidden states for the $T$ masks $\boldsymbol{h}_{y'_{mask_1}}, \ldots, \boldsymbol{h}_{y'_{mask_T}}$ as partial input to an auto-regressive Transformer-based decoder. Unlike generation tasks that require an `[EOS]` token to indicate the end of decoding, our decoder runs $T$ steps to predict the answer spans for these $T$ masks. At step $t$, we first fuse the hidden representation $\boldsymbol{h}_{\texttt{[MASK]}_t} \in \mathbb{R}^d$ of the $t$-th `[MASK]` token and previously predicted entity representation $\boldsymbol{s}_{t-1}^{ent} \in \mathbb{R}^d$:

$$\boldsymbol{z}_t = \mathbf{W}[\boldsymbol{h}_{\texttt{[MASK]}_t}; \boldsymbol{s}_{t-1}^{ent}]\,, \quad (8)$$

where $\mathbf{W} \in \mathbb{R}^{2d \times d}$, $\boldsymbol{s}_0^{ent} = \boldsymbol{h}_{\texttt{[CLS]}}$ (the representation of `[CLS]` token), and $[;]$ denotes vector concatenation.

The input $\boldsymbol{z}_t$ is then fed to the Transformer decoder (as in Eqn. (2) and (3)) to generate the decoder's hidden state $\boldsymbol{h}'_t$ at time step $t$. Based on $\boldsymbol{h}'_t$, we use a two-pointer network to predict the start and end positions of the answer entity in the source (encoder's hidden states). This is achieved with cross-attention of $\boldsymbol{h}'_t$ w.r.t. the encoder's hidden states, similar to Eqn (4) and (5). This operation results in two distributions over the encoder's hidden states for the start and end span positions. The final prediction of the start and end positions for mask $t$ is obtained by taking the argmax[4] over

[2] https://github.com/huggingface/transformers

[3] The segmentation embedding is used to distinguish the query (with two special tokens `[CLS]` and `[SEP]`) and the source in our models.

[4] The argmax is used for selecting the start and end indexes

the pointer position distributions:

$$p_{start} = \arg\max(a_1^{start}, ..., a_M^{start}), \quad (9)$$

$$p_{end} = \arg\max(a_1^{end}, ..., a_M^{end}), \quad (10)$$

under the constraint that $p_{start} < p_{end}$ and $p_{end} - p_{start} < k$.

Based on the start and end positions for the predicted entity, we can obtain the predicted entity representation at time step $t$ as the mean over the in-span encoder's hidden states:

$$\boldsymbol{s}_t^{ent} = \text{Mean-Pool}(\{\boldsymbol{h}_{p_{start}}, \boldsymbol{h}_{p_{end}}\}), \quad (11)$$

which is used as the input for the next step of decoding. It is worth noting that although the argmax operations in Eqn. (9) and (10) are non-differentiable, the model is trained based on the start and end positions of the ground-truth answer w.r.t. the start and end logits in Eqn. (4) and (5), which makes the gradient back-propagates to the encoder. Meanwhile, the encoder's hidden states used to compose $\boldsymbol{s}_i^{ent}$ in Eqn. (11) also carry the gradients. During inference, beam search is used to find the best sequence of predicted spans in the source to replace the masks.

Compared to the conventional Pointer Network (Vinyals et al., 2015; See et al., 2017) that only points to one token at a time, our sequential span selection decoder has the flexibility to replace a mask by any number of entity tokens, which is often required in summary factual correction.

## 4 Experiment

In this section, we present our results on using SpanFact for multiple summarization datasets.

### 4.1 Experimental Setup

Training data for our fact correction models are generated as described in Section 3.2 on CNN/DailyMail (Hermann et al., 2015), XSum (Narayan et al., 2018) and Gigaword (Graff et al., 2003; Rush et al., 2015). The statistics of these three dataset are provided in Table 2. During training, if an entity does not have a corresponding span in the source, we point the answer span to the `[CLS]` token. During inference, if the answer span predicted is the `[CLS]` token, we replace back the original masked entity.

for the answer span, and the softmax is used for computing the loss for back-propagation.

| Datasets | # docs (train/val/test) | doc len. | summ. len. | # mask |
|---|---|---|---|---|
| CNN | 90,266/1,220/1093 | 760.50 | 45.70 | 4.40 |
| DailyMail | 196,961/12,148/10,397 | 653.33 | 54.65 | 5.38 |
| XSum | 204,045/11,332/11,334 | 431.07 | 23.26 | 2.28 |
| Gigaword | 3,803,957/189,651/1,951 | 31.3 | 8.3 | 1.97 |

Table 2: Comparison of summarization datasets on train/validation/test set splits, average document and summary length (numbers of words). We also report the average number of entity masks on the reference summary for each dataset.

Our fact correction models are implemented via the Huggingface Transformers library (Wolf et al., 2019) in PyTorch (Paszke et al., 2017). We initialize all encoder models with the checkpoint of an uncased, large BERT model pre-trained on English data and SQuAD for all experiments. Both source and target texts were tokenized with BERT's sub-words tokenizer. The max sequence length is set to 512 for the encoder. We use a shallow Transformer decoder (L=2) for the auto-regressive span selection decoder, as the pre-trained BERT-large encoder is already robust for selecting right spans in the single-span selection task with only two pointers (Section 3.3). The Transformer decoder has 1024 hidden units and the feed-forward intermediate size for all layers is 4,096.

All models were finetuned on our span prediction data for 2 epochs with batch size 12. AdamW optimizer (Loshchilov and Hutter, 2017) with $\epsilon$ =1e-8 and an initial learning rate 3e-5 is used for training. Our learning rate schedule follows a linear decay scheduler with warmup=10,000. During inference, we use beam search with $b = 5$ and $k = 10$ (constraint for the distance between the start and end pointer). The best model checkpoints are chosen based on performance on the validation set. Experiments are conducted using 4 Quadro RTX 8000 GPUs with 48GB of memory.

### 4.2 Evaluation Metrics

We use three automatic evaluation metrics to evaluate our models. The first is ROUGE (Lin, 2004), the standard summarization quality metric, which has high correlation with summary informativeness in the news domain (Kryściński et al., 2019).

Since ROUGE has been criticized for its poor correlation with factual consistency (Kryściński et al., 2019; Wang et al., 2020), we use two additional automatic metrics that specifically focus on factual consistency: FactCC (Kryściński et al.,

| Datasets | QGQA | FactCC sent | ROUGE 1 | ROUGE 2 | ROUGE L |
|---|---|---|---|---|---|
| Bottom-up | 70.58 | 73.66 | **41.24** | **18.70** | **38.15** |
| Split Encoders | 70.22 | 73.15 | 39.78 | 17.87 | 37.01 |
| QA-Span | **74.15** | **76.60** | 41.13 | 18.58 | 38.04 |
| Auto-regressive | 72.78 | 74.42 | 41.04 | 18.48 | 37.95 |
| BertSumAbs | 72.68 | 76.76 | **41.67** | **19.46** | **38.79** |
| Split Encoders | 72.13 | 76.43 | 40.21 | 18.38 | 37.87 |
| QA-Span | **74.93** | **78.69** | 41.53 | 19.28 | 38.65 |
| Auto-regressive | 74.34 | 77.58 | 41.45 | 19.18 | 38.57 |
| BertSumExtAbs | 74.15 | 79.22 | **41.87** | **19.41** | **38.94** |
| Split Encoders | 73.67 | 79.12 | 40.55 | 18.41 | 38.45 |
| QA-Span | **75.94** | **80.97** | 41.75 | 19.27 | 38.81 |
| Auto-regressive | 75.19 | 79.89 | 41.68 | 19.16 | 38.74 |
| TransformerAbs | 73.79 | 80.51 | **39.96** | **17.63** | **36.90** |
| Split Encoders | 73.11 | 79.54 | 38.83 | 16.51 | 35.71 |
| QA-Span | **75.70** | **82.82** | 39.87 | 17.50 | 36.80 |
| Auto-regressive | 75.21 | 81.64 | 39.81 | 17.40 | 36.75 |

Table 3: Factual correctness scores and ROUGE scores on CNN/DailyMail test set.

| Datasets | QGQA | FactCC sent | ROUGE 1 | ROUGE 2 | ROUGE L |
|---|---|---|---|---|---|
| BertSumAbs | 12.78 | 23.60 | **37.78** | **15.84** | **30.50** |
| Split Encoders | **24.65** | 24.19 | 34.22 | 13.76 | 27.86 |
| QA-Span | 23.85 | 23.90 | 36.44 | 14.56 | 29.38 |
| Auto-regressive | 24.14 | **25.08** | 36.24 | 14.37 | 29.22 |
| BertSumExtAbs | 13.62 | 23.12 | **38.25** | **16.16** | **30.87** |
| Split Encoders | **25.17** | 24.67 | 35.66 | 13.98 | 27.93 |
| QA-Span | 24.52 | 23.96 | 36.86 | 14.82 | 29.70 |
| Auto-regressive | 24.96 | **25.10** | 36.67 | 14.64 | 29.53 |
| TransformerAbs | 7.00 | 24.15 | **29.86** | **10.05** | **23.78** |
| Split Encoders | 11.77 | 24.78 | 28.14 | 8.65 | 22.70 |
| QA-Span | 12.88 | 24.44 | 29.51 | 9.67 | 23.45 |
| Auto-regressive | **13.89** | **25.75** | 29.45 | 9.59 | 23.40 |

Table 4: Factual correctness scores and ROUGE scores on XSum test set.

2019) and QAGS (Wang et al., 2020). FactCC is a pre-trained binary classifier that evaluates the factuality of a system-generated summary by predicting whether it is consistent or inconsistent w.r.t. the source. This classifier was trained on adversarial examples obtained by heuristically injecting noise into reference summaries.

In addition, very recent work proposed QA-based models for factuality evaluation (Wang et al., 2020; Durmus et al., 2020; Maynez et al., 2020), and Wang et al. (2020) showed that their evaluation models have higher correlation with human judgements on factuality when compared with FactCC (Kryściński et al., 2019). We thus include our re-implementation of a question generation and question answering model (QGQA) following Wang et al. (2020) as an evaluation metric for factuality.[5] This model generates a set of questions based on the system-generated summary, and then answers these questions using either the source or the summary to obtain two sets of answers. The answers are compared against each other using an answer-similarity metric (token-level F1), and the averaged similarity metric over all questions is used as the QGQA score. Answers generated from a highly faithful system summary should be similar to those generated from the source.

### 4.3 Baselines

We compare against the following abstractive summarization baselines. On CNNDM and XSum, we use BertSumAbs, BertSumExtAbs and TransformerAbs (Liu and Lapata, 2019). In addition, we also compare with Bottom-up (Gehrmann et al., 2018). On Gigaword, we use the pointer-generator (See et al., 2017), base and full GenParse models (Song et al., 2020) for comparison. For the factual correction baseline, we compare with the Two-encoder Pointer Generator[6] (Split Encoder) (Shah et al., 2020), which employs a similar setting to ours for masking entities w.r.t. the source, and uses dual encoders to copy and generate from both the source and the masked query for fact update. Compared to our span selection models that can fill in the mask with any number of tokens, their models aim to regenerate the mask query based on the source. In other words, their decoder regenerates the whole sequence token by token with a pointer-generator, which inherits the backbone of generative models that suffer from hallucination.

### 4.4 Experimental Results

Tables 3, 4, and 5 summarize the results on the CNN/DailyMail, XSum and Gigaword datasets, respectively. Each block in the tables compares the original summarization model's output with

[5]We were not able to obtain any of the QA evaluation model or code from Wang et al. (2020); Durmus et al. (2020); Maynez et al. (2020) as the authors are still in the stage of making the code public. We used pre-trained UniLM model for question generation (QG) and BertForQuestionAnswering model for question answering (QA). The QG model is fine-tuned on NewsQA (Trischler et al., 2017) with entity-answer conditional task (Wang et al., 2020), and the QA model is pre-trained on SQuAD 2.0 (Rajpurkar et al., 2018).

[6]https://github.com/darsh10/split_encoder_pointer_summarizer

| Datasets | QGQA | FactCC sent | ROUGE 1 | 2 | L |
|---|---|---|---|---|---|
| GenParse (base) | 52.63 | 46.07 | **35.23** | **17.11** | **32.88** |
| Split Encoders | 63.60 | 48.22 | 34.32 | 17.01 | 31.98 |
| QA-Span | **66.47** | **52.17** | 34.38 | 16.50 | 32.07 |
| Auto-regressive | 64.77 | 48.95 | 33.97 | 16.08 | 31.70 |
| GenParse (full) | 55.47 | 48.44 | **36.61** | **18.85** | **34.32** |
| Split Encoders | 65.88 | 52.11 | 35.01 | 17.54 | 32.96 |
| QA-Span | **67.12** | **54.59** | 35.66 | 18.01 | 33.35 |
| Auto-regressive | 66.48 | 52.18 | 35.04 | 17.27 | 32.75 |
| Pointer Generator | 45.98 | 43.62 | **34.19** | **16.29** | **31.81** |
| Split Encoders | 59.46 | 48.32 | 33.11 | 15.63 | 30.67 |
| QA-Span | 58.25 | 45.62 | 33.30 | 15.70 | 30.95 |
| Auto-regressive | **60.66** | **49.82** | 32.86 | 15.22 | 30.51 |

Table 5: Factual correctness scores and ROUGE scores on Gigaword test set.

| BertAbs | Better | Worse | Same |
|---|---|---|---|
| QA-Span vs. original | **28.6%** | 18.7% | 52.7% |
| Auto-regressive vs. original | **31.3%** | 16.7% | 52% |
| QA-Span vs. Auto-regressive | 26% | 27.3% | 46.7% |
| TransformerAbs | Better | Worse | Same |
| QA-Span vs. original | **38%** | 11.3% | 40.7% |
| Auto-regressive vs. original | **36%** | 19.3% | 44.7% |
| QA-Span vs. Auto-regressive | 32.7% | 28% | 39.3% |
| Bottom-up | Better | Worse | Same |
| QA-Span vs. original | **34%** | 12% | 54% |
| Auto-regressive vs. original | **31.4%** | 13.3% | 55.3% |
| QA-Span vs. Auto-regressive | 41.3% | 32% | 26.7% |

Table 6: Human evaluation results on pairwise comparison of factual correctness on 450 ($9 \times 50$) randomly sampled articles.

the corrected outputs obtained by our baseline and proposed models.

On CNN/DailyMail (Table 3), our correction models significantly boost factual consistency measures (QGQA and FactCC) by large margins, with only small drops on ROUGE. This shows our models have the ability to improve the correctness of system-generated summaries without sacrificing informativeness. When comparing our two proposed models, we observe that the QA-span model performs better than the auto-regressive model. This is expected as CNN/DailyMail's reference summaries tend to be more extractive (See et al., 2017), and summarization models tend to make few errors per summary (Narayan et al., 2018). Thus, the iterative procedure of the QA-span model is more robust with high precision as it has more correct context from the query, with only minimum negative influence from other concurrent errors. This is also reflected in the high scores of QGQA and FactCC across all the models we tested. Since QGQA and FactCC are based on comparing system-generated summary w.r.t. the source text, high score means high semantic similarity between system summary to the source.

On XSum (Table 4) and Gigaword (Table 5), both of our correction models boost factual consistency measures by large margins with a slight drop in ROUGE (-0.5 to -1.5) on average. This is still encouraging, as abstractive summarization models that use complex factual controlling components for generation often have drops of 5-10 ROUGE points (Zhu et al., 2020).

We also notice that the QGQA and FactCC scores of all summarization models are lower than that on CNN/DailyMail. The scores are especially low on XSum. This is likely due to the data construction protocol of XSum, where the first sentence of a source document is used as the summary and the remainder of the article is used as the source. As a result, many entities that appear in the reference summary never appear in the source, which may cause abstractive summarization models to hallucinate severely with many factual errors (Maynez et al., 2020). As the system summaries often contain many errors, our QA-span model that relies on answering a single-mask query often has the wrong context to condition on at each step, which negatively affects the performance of this model. In contrast, the strategy of masking all the entities would provide the auto-regressive model a better query for entity replacement. We can observe in Table 4 that the auto-regressive model performs better than the QA-span model on XSum.

### 4.5 Human Evaluation

To provide qualitative analysis of the proposed models, we conduct human evaluation on pairwise comparison of CNN/DailyMail summaries enhanced by different correction strategies. We select three state-of-the-art abstractive summarization models as the backbones, and collect three sets of pairwise summaries for each setting: $(i)$ Original vs. QA-Span corrected; $(ii)$ Original vs. Auto-regressive corrected; $(iii)$ QA-Span corrected vs. Auto-regressive corrected. Nine sets of 50 randomly selected samples (total 450 samples) are labeled by AMT tuckers. For each pair (in anonymized order), three annotators from Amazon Mechanical Turk (AMT) are asked to judge which is more factually correct based on the

| FactCC Dataset | FactCC Score | QAQG | Human Eval |
|---|---|---|---|
| Before Corr. | 84.89 | 88.65 | 87.79 |
| QA-span | **86.08** | **91.07** | **90.74** |
| Auto-regressive | 85.96 | 90.51 | 90.35 |

Table 7: Test results on the human annotated dataset provided by FactCC (Kryściński et al., 2019). We show the performance comparisons of the original summaries and the summaries corrected by SpanFact.

source document. As shown in Table 6, summaries from our two models are chosen more frequently as the factually correct one compared to the original. Between the two correction models, the preferences are comparable.

In addition, we also test our fact correction models on the FactCC test set provided by Kryściński et al. (2019) and manually checked the outputs. Table 7 shows the results of the original summaries and the summaries corrected by our models in terms of automatic fact evaluation and our manual evaluation. Among 508 system-generated summary sentences, 62 were incorrect. The QA-span model was able to correct 18 out of 62 right, and the auto-regressive model was able to correct 16 out of 62. Among the 446 sentences that are labeled as correct by the annotators in Kryściński et al. (2019), our two models made 3 and 4 wrong changes in the entities, respectively,[7] while keeping most of the entities unchanged or changed with equivalent entities.

## 5 Conclusion

We present SpanFact, a suite of two factual correction models that use span selection mechanisms to replace one or multiple entity masks at a time. SpanFact can be used for fact correction on any abstractive summaries. Empirical results show that our models improve the factuality of summaries generated by state-of-the-art abstractive summarization systems without a huge drop on ROUGE scores. For future work, we plan to apply our method for other type of spans, such as noun phrases, verbs, and clauses.

## Acknowledgments

This research was supported in part by Microsoft Dynamics 365 AI Research and the Canada CIFAR AI Chair program. We would like to thank the reviewers for their valuable comments and special thanks to Yuwei Fang and other members of the Microsoft Dynamics 365 AI Research team for the feedback and suggestions.

[7] This excludes the cases where the model would change a person's full name by last name or break the fluency due to SpaCy NER errors.

## References

Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2018. Faithful to the original: Fact aware neural abstractive summarization. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Sumit Chopra, Michael Auli, and Alexander M Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–98.

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.

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*.

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

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

Sebastian Gehrmann, Yuntian Deng, and Alexander M Rush. 2018. Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4098–4109.

Ben Goodrich, Vinay Rao, Peter J Liu, and Mohammad Saleh. 2019. Assessing the factual accuracy of generated text. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 166–175.

David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2003. English gigaword. *Linguistic Data Consortium, Philadelphia*, 4(1):34.

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

Matthew Honnibal and Ines Montani. 2017. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*, 7(1).

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39.

Wojciech Kryściński, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Evaluating the factual consistency of abstractive text summarization. *arXiv preprint arXiv:1910.12840*.

Katherine Lee, Orhan Firat, Ashish Agarwal, Clara Fannjiang, and David Sussillo. 2018. Hallucinations in neural machine translation.

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*.

Haoran Li, Junnan Zhu, Jiajun Zhang, and Chengqing Zong. 2018. Ensure the correctness of the summary: Incorporate entailment knowledge into abstractive sentence summarization. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1430–1441.

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

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. 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 3721–3731.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

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

Cao Meng, Yue Cheung Dong, Jiapeng Wu, and Jackie Chi Kit. 2020. Factual error correction for abstractive summarization models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

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

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

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch.

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *International Conference on Learning Representations*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

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.

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.

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

Darsh J Shah, Tal Schuster, and Regina Barzilay. 2020. Automatic fact-guided sentence modification. In *Proceedings of The Thirty-Fourth AAAI Conference on Artificial Intelligence*.

Tian Shi, Ping Wang, and Chandan K. Reddy. 2019. LeafNATS: An open-source toolkit and live demo system for neural abstractive text summarization. In *Proceedings of the 2019 Conference of the North*

*American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 66–71, Minneapolis, Minnesota. Association for Computational Linguistics.

Kaiqiang Song, Logan Lebanoff, Qipeng Guo, Xipeng Qiu, Xiangyang Xue, Chen Li, Dong Yu, and Fei Liu. 2020. Joint parsing and generation for abstractive summarization. In *Proceedings of The Thirty-Fourth AAAI Conference on Artificial Intelligence*, pages 4098–4109.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. Newsqa: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in neural information processing systems*, pages 2692–2700.

Oriol Vinyals and Quoc Le. 2015. A neural conversational model. *arXiv preprint arXiv:1506.05869*.

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*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J Liu. 2020a. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *Thirty-seventh International Conference on Machine Learning (ICML 2020)*.

Tianyi Zhang, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020b. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Chenguang Zhu, William Hinthorn, Ruochen Xu, Qingkai Zeng, Michael Zeng, Xuedong Huang, and Meng Jiang. 2020. Boosting factual correctness of abstractive summarization with knowledge graph. *arXiv preprint arXiv:2003.08612*.

| | |
|---|---|
| CNNDM Source | Jerusalem (CNN)The flame of remembrance burns in Jerusalem, and a song of memory haunts Valerie Braham as it never has before. This year, Israel's Memorial Day commemoration is for bereaved family members such as Braham. "Now I truly understand everyone who has lost a loved one," Braham said. Her husband, Philippe Braham, was one of 17 people killed in January's terror attacks in Paris. He was in a kosher supermarket when a gunman stormed in, killing four people, all of them Jewish. |
| System Summary | france's memorial day commemoration is for bereaved family members as braham. valerie braham was one of 17 people killed in january's terror attacks in paris. |
| Corrected by SpanFact | israel's memorial day commemoration is for bereaved family members as braham. philippe braham was one of 17 people killed in january's terror attacks in paris. |
| CNNDM Source | (CNN)If I had to describe the U.S.-Iranian relationship in one word it would be "overmatched." ... America is alienating some of our closest allies because of the Iran deal, and Iran is picking up new ones and bolstering relations with old ones who are growing more dependent because they see Iranś power rising... |
| System Summary | iran is alienating some of our closest allies because of the iran deal, and iran is picking up new ones. |
| Corrected by SpanFact | america is alienating some of our closest allies because of the iran deal, and iran is picking up new ones. |
| CNNDM Source | (CNN)A North Pacific gray whale has earned a spot in the record books after completing the longest migration of a mammal ever recorded. The whale, named Varvara, swam nearly 14,000 miles (22,500 kilometers), according to a release from Oregon State University, whose scientists helped conduct the whale-tracking study. Varvara, which is Russian for "Barbara," left her primary feeding ground off Russiaś Sakhalin Island to cross the Pacific Ocean and down the West Coast of the United States to Baja, Mexico... |
| System Summary | a north pacific gray whale swam nearly 14,000 miles from oregon state university. |
| Corrected by SpanFact | a north pacific gray whale swam nearly 14,000 miles from russiaś sakhalin island. |
| CNNDM Source | Sanaa, Yemen (CNN)Saudi airstrikes over Yemen have resumed once again, two days after Saudi Arabia announced the end of its air campaign. The airstrikes Thursday targeted rebel Houthi militant positions in three parts of Sanaa, two Yemeni Defense Ministry officials said. The attacks lasted four hours. ... The Saudi-led coalition said a new initiative was underway, Operation Renewal of Hope, focused on the political process. But less than 24 hours later, after rebel forces attacked a Yemeni military brigade, the airstrikes resumed, security sources in Taiz said. |
| System Summary | the attacks lasted four hours, two days after rebel forces attacked yemeni military troops.. |
| Corrected by SpanFact | the attacks lasted four hours, less than 24 hours after rebel forces attacked yemeni military troops. |
| CNNDM Source | Boston (CNN)When the bomb went off, Steve Woolfenden thought he was still standing. That was because, as he lay on the ground, he was still holding the handles of his son's stroller. He pulled back the stroller's cover and saw that his son, Leo, 3, was conscious but bleeding from the left side of his head. Woolfenden checked Leo for other injuries and thought, "Let's get out of here." ... |
| System Summary | steve woolfenden, 3, was conscious but bleeding from the left side of his head. |
| Corrected by SpanFact | leo, 3, was conscious but bleeding from the left side of his head. |

Table 8: Examples of factual error correction on FactCC dataset (a human annotated subset from CNNDM obtained by Kryściński et al. (2019)). Factual errors by abstractive summarization system are marked in red. Corrections made by the proposed SpanFact models are marked in orange.