Entity-based SpanCopy for Abstractive Summarization to Improve the Factual Consistency

Anonymous ACL submission

Abstract

Despite the success of recent abstractive summarizers on automatic evaluation metrics, the generated summaries still present factual inconsistencies with the source document. In this paper, we focus on entity-level factual inconsistency, i.e. reducing the mismatched entities between the generated summaries and the source documents. We therefore propose a novel entity-based SpanCopy mechanism, and explore its extension with a Global Relevance component . Experiment results on four summarization datasets show that SpanCopy can effectively improve the entity-level factual consistency with essentially no change in the word-level and entity-level saliency. [1]

1 Introduction

Abstractive text summarization, the task to generating informative and fluent summaries of the given document(s), has attracted much attention in the NLP community. While early neural approaches focused more on designing customized architectures or training schema to better fit the summarization task (Nallapati et al., 2016; Tan et al., 2017; Liu* et al., 2018), recent works have shown that generation models, pre-trained on large corpora (Lewis et al., 2020; Zhang et al., 2020; Raffel et al., 2020), generally have a better performance when fine-tuned on in-domain datasets.

However, even if these pre-trained&fine-tuned generation models achieve state-of-the-art performance with respect to standard automatic evaluation metrics, e.g. ROUGE score(Lin, 2004) and BERTScore(Zhang* et al., 2020), the generated summaries still suffer from the problem of factual inconsistency, which means the generated summaries may not be factually consistent with the content expressed in the source documents (Kryscinski et al., 2020). Inconsistencies may exist either at the entity or the relation level (Nan et al., 2021). The former case is when the summary mentions an *entity* that does not appear in the source documents. The latter is when the summary does mention entities from the source documents, but expresses a *relation* between them which is different than the one stated in the source documents.

In this paper, we focus on the entity-level inconsistency problem, i.e. to make the model generate summaries with less entities which do not appear in the source document(s) i.e., 'hallucinated' entities. Note however, that hallucinated entities are not necessarily 'unfaithful' or 'wrong'(Cao et al., 2021), so the goal is to reduce them without excluding entities that do appear in the reference summary i.e., without penalizing saliency. Table 1 shows an example of entity-level factual inconsistency from the XSum dataset. Although the content of the summary generated by the SOTA summarizer PEGASUS (Zhang et al., 2020) is roughly similar that of the ground-truth summary, it does not accurately summarize the original documents with the *proper entities*. Specifically, it totally misses the entity 'Royal Marine', which appears in both the source document and the reference summary, and the entity 'Hampshire' is 'hallucinated', as it does not appear in the source document. Despite the fact that the city 'Portsmouth' is located in 'Hampshire' county, the entity itself is still an instance of factual inconsistency (i.e., an unnecessary generalization).

Prior work (Dong et al., 2020; King et al., 2022) mainly address the entity-level inconsistency problem in the post-processing stage. However, those methods either requires additional sophisticated models, e.g. Dong et al. (2020) uses a pre-trained QA model to 'revise' the generated summaries, or being built on arguably brittle heuristics (King et al., 2022) . Recent work (Nan et al., 2021) proposes two ways to directly improve the end-to-end summarization model, either by training with an auxiliary task, which is to recognize the summary-

[1] The code will be published in the final version.

Entities in Source Doc: Royal Marine, Falklands, Portsmouth, Falklands War Memorial....
Ground Truth: Plans to move a statue depicting a Royal Marine in the Falklands conflict away from Portsmouth seafront have been criticised.
PEGASUS: A campaign has been launched to keep a statue of a Falklands War marine in Hampshire.
SpanCopy: A campaign to keep a statue of a Royal Marine marching across the Falklands in Portsmouth has been launched.
SpanCopy + GR: A statue of a Royal Marine marching across the Falklands during the Falklands War Memorial should remain in its current location, campaigners have said.

Table 1: An example of entity-level factual inconsistency from the XSum dataset. The summary generated by PEGASUS totally missed one entity (Royal Marine) and one entity indicates a larger area than the correct one (Hampshire).

worthy entities in the source document using the hidden states from the encoder, or jointly generating the entities and the summaries, i.e. generating a chain of entities in the summary followed by the summary. Yet, both methods do not explicitly encourage the model to generate the summaries with more valuable entities, as both of them aim to guide the model to detect the summary-worthy entities without any changes in the summary generation process. Instead, aiming for a lean and modular solution, we propose the SpanCopy Mechanism to explicitly copy the matched entities[2] from the source documents when generating the summaries. One key advantage of our proposal is that it can be easily integrated into any pre-trained generative sequence-to-sequence model.

Since often only a few of the entities in the source documents can be included in the summary, which we call 'summary-worthy entities', we also explore an additional Global Relevance component to better recognize the summary-worthy entities by automatically generating a prior distribution over all the entities in the source documents.

We test our proposal on four summarization datasets in the news and scientific paper domain, comparing it with the SOTA PEGASUS system (Zhang et al., 2020). In a first set of experiments, as a sanity check, we assess our models on arguably easier subsets of these datasets, where all the entities in the reference summaries belong to the source document. In these cases, SpanCopy should definitely dominate PEGASUS, which is confirmed by the results. In a second set of experiments, we fine-tune and test on the full datasets. On this realistic and more challenging task, we find that SpanCopy (without Global Relevance) can strongly improve the entity-level factual consistency (+2.28) on average across datasets, with essentially no change in saliency (−0.06).

2 Related Work

2.1 Abstractive Summarization

Early neural abstractive summarization models (Nallapati et al., 2016; Paulus et al., 2018; See et al., 2017) are mainly sequence-to-sequence models based on different variants of RNN, e.g. LSTM or GRU, with additional components targeting different properties of the summaries, like redundancy (Tan et al., 2017) and coverage (See et al., 2017). However, all the recurrent models suffer from serious weakness like long-term memory loss, and requiring excessive time to train.

To tackle these problems, researchers in the area of abstractive summarization started to use attention-based transformer models (Liu and Lapata, 2019a,b); recently reaching SOTA performance when pre-trained generative transformers are applied to the task, e.g. BART (Lewis et al., 2020), PEGASUS (Zhang et al., 2020) and PRIMERA (Xiao et al., 2021). The SpanCopy mechanism we propose in this paper can be advantageously injected into any pre-trained models.

2.2 Factual Consistency

Despite the large improvements with respect to automatic evaluation metrics, recent studies (Cao et al., 2018; Kryscinski et al., 2020) show that around 30% of the summaries generated by the SOTA summarization models contain factual inconsistencies. Ideally, the assessment of factual consistency should rely on human annotations (Maynez et al., 2020), but these are costly, time consuming and lack a unified standard. Thus promising automatic evaluation metrics for factual consistencies of generated summaries have been explored in recent years. To assess relation-level factual consis-

[2] We particularly focus on the Named Entities in this paper, but our method can be easily applied to any kinds of span or entities.

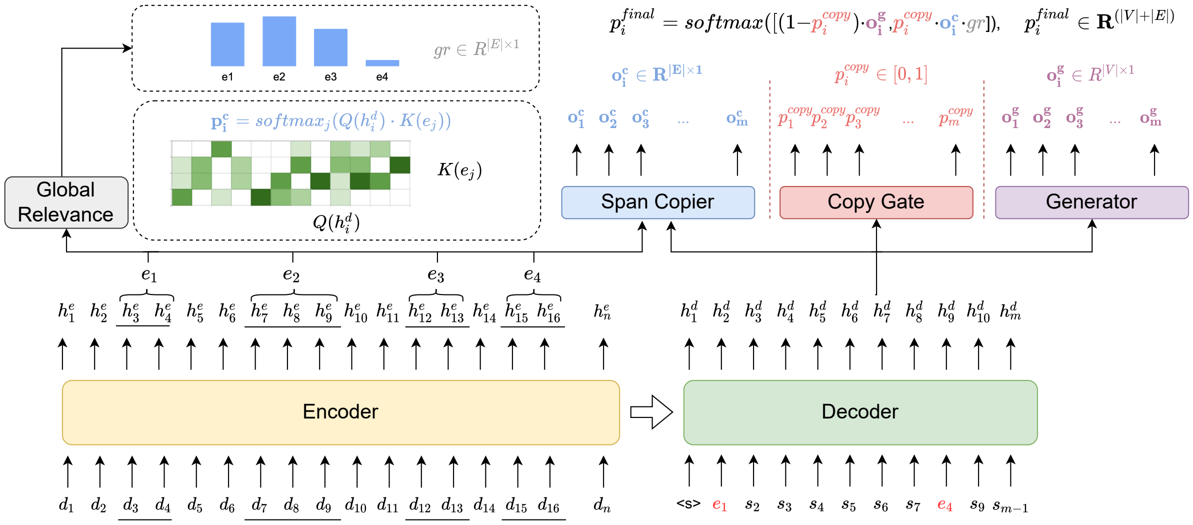
$gr \in R^{|E|\times 1}$
e1
e2
e3
e4
$\mathbf{p}_i^c = softmax_j(Q(h_i^d) \cdot K(e_j))$
$K(e_j)$
$Q(h_i^d)$
Global Relevance
$p_i^{final} = softmax([(1-p_i^{copy})\cdot\mathbf{o}_i^g, p_i^{copy}\cdot\mathbf{o}_i^c\cdot gr]), \quad p_i^{final} \in \mathbf{R}^{(|V|+|E|)}$
$\mathbf{o}_i^c \in \mathbf{R}^{|E|\times 1}$
$p_i^{copy} \in [0,1]$
$\mathbf{o}_i^g \in R^{|V|\times 1}$
$\mathbf{o}_1^c$ $\mathbf{o}_2^c$ $\mathbf{o}_3^c$ ... $\mathbf{o}_m^c$
p_1^{copy} p_2^{copy} p_3^{copy} ... p_m^{copy}
$\mathbf{o}_1^g$ $\mathbf{o}_2^g$ $\mathbf{o}_3^g$ ... $\mathbf{o}_m^g$
Span Copier
Copy Gate
Generator
e_1 e_2 e_3 e_4
h_1^e h_2^e h_3^e h_4^e h_5^e h_6^e h_7^e h_8^e h_9^e h_{10}^e h_{11}^e h_{12}^e h_{13}^e h_{14}^e h_{15}^e h_{16}^e h_n^e
h_1^d h_2^d h_3^d h_4^d h_5^d h_6^d h_7^d h_8^d h_9^d h_{10}^d h_m^d
Encoder
Decoder
d_1 d_2 d_3 d_4 d_5 d_6 d_7 d_8 d_9 d_{10} d_{11} d_{12} d_{13} d_{14} d_{15} d_{16} d_n
<s> e_1 s_2 s_3 s_4 s_5 s_6 s_7 e_4 s_9 s_{m-1}

Figure 1: Structure of the model with Entity-based SpanCopy Mechanism, with five components: Encoder, Decoder, Span Copier, Copy Gate and Generator. The upper left bar plot shows the Global Relevance component, predicting the prior probability of all the entities $\{e_1, e_2, e_3, e_4\}$ to be copied to the summary.

tency two kinds of metrics have been proposed: one based on classification (Kryscinski et al., 2020), and one based on Question-Answering (Maynez et al., 2020; Durmus et al., 2020). For entity-level factual consistency, the focus of this paper, Nan et al. (2021) propose a simple but effective evaluation metric, based on the matched named entities in both generated and ground-truth summaries. In our work, we use such metric to evaluate whether the generated summaries are consistent with both the source documents and the reference summaries at the entity-level.

2.3 Copy Mechanism

See et al. (2017) first apply pointer-generator network in an abstractive summarization model, which facilitates copying words from the source documents by pointing, i.e., generating a distribution of probabilities to copy each word from the source. Following their work, Bi et al. (2020) propose PALM, in which the copy mechanism is applied on top of the transformer model, and with a novel pre-training schema, the model achieves SOTA on several generative tasks, such as abstractive summarization and generative QA. More recently, Li et al. (2021) further explores how to make use of the copy history to predict the copy distribution for the current step. However, all the aforementioned works focus on copying at the word level, which tends to be sparse and noisy. Instead, we aim to train the model to copy spans of text i.e., the named entities, in this paper.

Admittedly, some previous work has also investigated span-based copy mechanisms. Yet, those models either predict the start and end indices of a span (Zhou et al., 2018), or predict the BIO labels for each token (Liu et al., 2021). Even if such strategies can copy any kinds of spans (clauses, n-grams, entities, phrases or longest common sequence) from the source document, they may introduce unnecessary noise and break the coherence of the generated text. In this work, we focus on copying the spans of the Named Entities, extracted by a high-quality NER tool, aiming to improve factual consistency of the generated summary without negatively affecting saliency.

3 Our SpanCopy Method

3.1 Transformer-based Summarizers

Typically, transformer-based summarization(Lewis et al., 2020; Zhang et al., 2020) consists of two steps (i) The **Encoding Step** (by the Encoder shown in yellow in Fig.1), which encodes the source input(s) into an hidden space; (ii) the **Decoding Step**, which computes a probability distributions on the output vocabulary to generate each token of the resulting summary. In this paper, to better describe our methods in the context of a generic summarization models, we split the Decoding process into two components, the Decoder itself (shown in green in Fig.1), which outputs the representations of predicted tokens, and the Generator (shown in purple in Fig.1), an MLP layer mapping the representations to the final probability distribution on the output vocabulary.

More formally, for a document with n tokens $D = \{t^d_1, t^d_2, ..., t^d_n\}$, and the corresponding summary with m tokens, $S = \{t^s_1, t^s_2, ..., t^s_m\}$, the output of the Encoder is a sequence of hidden states of all the tokens, i.e. $\{h^e_1, h^e_2, ..., h^e_n\}$. And then the Decoder predicts a sequence of vector, $\{h^d_1, h^d_2, ..., h^d_m\}$, representing the tokens to be predicted. Finally, the Generator maps those vectors to the distributions over the vocabulary, i.e. $\{\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_m\}$, where $\mathbf{p}_i \in R^{|V|}$.

3.2 SpanCopy Mechanism

A key problem with generic sequence-to-sequence transformer-based summarizers is that the decoding step is prone to generate factual inconsistencies, i.e. the model may make up entities or relations that are not entailed by the source documents. To address entity-level factual inconsistency, we introduce in the Decoding Step the SpanCopy mechanism, which can be conveniently plugged into any pre-trained models. Specifically, we first identify and match the entities in both source document and summary, and then instead of generating the entire summary word by word, we add an additional Span Copier to directly copy entities from the source document, with a Copy Gate predicting the likelihood of whether the model should generate the current token from the vocabulary or directly copy an entity from the source document.

Span Copier (shown in blue in Fig.1) is an attention module over all the entities in the input document. Suppose there are $|E|$ entities in the input document, with each entity j being a span over tokens $[d_{j_s}, d_{j_e}]$, then the entities can be simply represented as $e_j = \mathbf{avg}([h^e_{j_s} : h^e_{j_e}])$, where h^e_i represents the output of the encoder for each token d_i. At each decoding step i, we compute the logit vector of copying each entity at the current step as:

$$\mathbf{o^c_i} = Q(h^d_i) \cdot K(e_j), \mathbf{o^c_i} \in \mathbf{R}^{|E|} \quad (1)$$

indicating how likely it is to copy the entities from the source document at each step. Notice that to better balance the numeric difference caused by the size of selection space ($|V|$ and $|E|$), we generate and combine the raw logit vectors[3] from the Span Copier and Generator, and take softmax over the combined space to get the final probability.

Copy Gate (shown in red in Fig.1) is a classifier to map the hidden states to a singular value, i.e.

$$p^{copy}_i = \sigma(MLP(h^d_i)), p^{copy}_i \in [0, 1] \quad (2)$$

which indicates the probability of copying an entity at each step. On the contrary, $1 - p^{copy}_i$ represent the probability of generating a token from the vocabulary at step i.

Then the final probability, combining both generation over the vocabulary and the copy mechanism over the entity space, is computed as

$$\mathbf{p^{final}_i} = softmax([(1 - p^{copy}_i) \cdot \mathbf{o^g_i}, p^{copy}_i \cdot \mathbf{o^c_i}]) \quad (3)$$

with $\mathbf{p^{final}_i} \in \mathbf{R}^{(|V|+|E|)}$, where $\mathbf{o^g_i} \in \mathbf{R}^{(|V|)}$ is the logit vector of token generation and $\mathbf{o^c_i} \in \mathbf{R}^{(|E|)}$ is the logit vector of entity copying. As a result, the first $|V|$ dimensions of the final probability represent the probability of generating all the tokens from the vocabulary, while the following $|E|$ dimensions contain the probabilities of copying the entities from the source document.

Note that the input of the original Decoder in the transformer model at each step is the embedding of the previous token (which is the ground-truth one during training, and the predicted one for inference), but a span of text longer than 1 does not naturally have an embedding to match. We simply use the average of the embedding of all the tokens in the entity, following previous work using average embedding to represent a span of text (Xiao and Carenini, 2019).

3.3 Loss

We use the standard loss for abstractive summarization, i.e. the cross entropy loss between the predicted probability and the ground truth labels,

$$L_1 = \sum_i L_s(\mathbf{p^{final}_i}, t_i) \quad (4)$$

However, notice that, since the predicted probability distribution is over the combined space of vocabulary size and entity size ($\mathbf{p^{final}_i} \in \mathbf{R}^{|V|+|E|}$), the corresponding ground truth labels can be either indices of words to be generated from the vocabulary, or the indices of entities to be copied from the source document, i.e. $t_i \in [0, |V| + |E|]$. Specifically, if $t_i < |V|$, then the t_i-th token should be generated, and if $t_i > |V|$, the $(t_i - |V|)$-th entity should be copied from the source document.

[3] The vector of raw (non-normalized) predictions that the classification model generates

Dataset	Original					Filtered				
	L_{doc}	L_{summ}	N_{doc}	N_{summ}	$src_p(gt)$	L_{doc}	L_{summ}	N_{doc}	N_{summ}	$src_p(gt)$
CNNDM	690.9	52.0	42.8	5.9	80.41	671.9	47.1	39.4	4.4	100
XSum	373.8	21.1	27.9	2.7	39.85	483.4	20.6	31.6	1.9	100
Pubmed	3049.0	202.4	71.1	6.4	70.93	3165.4	178.5	69.9	3.4	100
arXiv	6033.3	271.5	157.5	6.0	39.12	6478.9	164.1	161.9	2.3	100

Table 2: Statistics of all the datasets (original/filtered), on the lengths (L_{doc},L_{summ}) and number of entities (N_{doc}, N_{summ}) in the source documents and ground truth summaries, as well as $src_p(gt)$, the entity level source-precision of the ground-truth summary.

3.4 SpanCopy with Global Relevance

Among all the entities in the source documents, there are only a few summary-worthy entities that should be copied into the summary (e.g. around 10% in CNNDM and 1.5% in arXiv). To make the model better recognize such summary-worthy entities, we explore a Global Relevance (GR) component, which takes all the entities in the source document as inputs, and predicts how likely each entity is to appear in the final summary. We use the generated 'entity likelihood' as a prior distribution for the Span Copier component, with GR also trained as an auxiliary task.

Global Relevance is a classifier mapping the hidden state of a source document entity into a value within $[0, 1]$, indicating the probability that such entity should be included in the summary.

$$\mathbf{gr} = \sigma(MLP(\mathbf{e})), \mathbf{gr} \in \mathbf{R}^{|E|} \tag{5}$$

Then p_i^{final} in Eq.3 is updated with gr as

$$\begin{aligned} \mathbf{p_i^{final}} = softmax([(1 - p_i^{copy}) \cdot \mathbf{o_i^g} \\ , p_i^{copy} \cdot \mathbf{o_i^c} \cdot \mathbf{gr}]) \end{aligned} \tag{6}$$

New Loss As an auxiliary task, we also train the model with the ground-truth GR labels to make it more accurate. Specifically, the label $y_i^{gr} = 1$ if the i-th entity in the input document is included in the ground truth summary. Then we update the loss function with L_{gr} balanced by β:

$$\begin{aligned} L_2 =& (1 - \beta) \sum_i L_s(\mathbf{p_i^{final}}, t_i) \\ &+ \beta \sum_j L_{gr}(gr_j, y_j^{gr}) \end{aligned} \tag{7}$$

4 Experiments and Analysis

4.1 Settings

SpanCopy can be plugged into any pre-trained generation model. In this paper, we use PEGASUS(Zhang et al., 2020) as our base model, since it has delivered top performance on multiple summarization datasets. We recognize named entities with an off-the-shelf NER tool[4]. The balance factor β of GR is set by grid search on small subsets of each dataset (2k for training and 200 for validation).

Dataset	# Data (original)	# Data (filtered)
CNNDM	287,113/13,368/13,368	105,847/4,490/3,903
XSum	204,017/11,327/11,333	42,481/2,349/2,412
Pubmed	119,924/6,633/6,658	32,123/1,797/1,772
arXiv	202,914/6,436/6,440	66,360/2,365/2,324

Table 3: Number of data examples in all the datasets (original v.s. filtered).

4.2 Evaluation Metrics

To evaluate the saliency and entity-level factual consistency of the generated summaries, we apply the following metrics:

Saliency metrics assess the similarity of the generated summary with the reference summary.

ROUGE scores (Lin, 2004) measure the n-gram overlaps between generated and ground truth summaries. We apply the metrics R-1, R-2 and R-L.

Summary-precision, -recall and -f1 (sum_p, sum_r and sum_f) (Nan et al., 2021) measure the precision/recall/f1 score of the matched entities in the generated summaries and the reference summaries. we use $NE(S_{ref})$ and $NE(S_{gen})$ to represent the named entities in the reference summaries and generated summaries, respectively.

$$\begin{aligned} sum_p &= |NE(S_{ref}) \cap NE(S_{gen})| / |NE(S_{gen})| \\ sum_r &= |NE(S_{ref}) \cap NE(S_{gen})| / |NE(S_{ref})| \\ sum_f &= 2 * (sum_p + sum_r) / sum_p * sum_r \end{aligned}$$

These three metrics measure the entity-level saliency of the generated summaries, i.e. recognizing how many copied (and generated) entities are salient, and should be included in the summary.

[4] https://spacy.io/

Model	ROUGE			Entity(Summ)			Entity(Doc)
	R-1	R-2	R-L	sum_r	sum_p	sum_f	src_p
CNNDM Filtered							
PEGASUS	44.70	22.23	32.52	50.80	45.32	45.03	92.85
SpanCopy	45.46	23.12	33.48	53.08	48.63	47.86	94.64
SpanCopy+GR	45.74	23.44	33.67	54.61	48.27	48.36	95.02
XSum Filtered							
PEGASUS	43.01	19.00	34.01	59.14	54.94	54.68	77.32
SpanCopy	44.23	19.90	35.50	61.34	59.15	58.16	84.30
SpanCopy+GR	43.78	19.12	34.97	60.69	60.50	58.36	83.75
Pubmed Filtered							
PEGASUS	46.99	21.46	42.57	42.63	33.28	33.16	73.59
SpanCopy	47.82	22.34	43.43	41.58	34.12	33.44	73.74
SpanCopy+GR	48.04	22.18	43.56	42.11	36.21	34.86	74.15
arXiv Filtered							
PEGASUS	46.23	18.02	41.02	37.65	35.98	33.48	68.13
SpanCopy	46.36	18.29	41.23	39.50	37.61	34.95	72.12
SpanCopy+GR	46.56	18.27	41.34	35.38	36.11	32.76	67.56

Table 4: Result of our models and the compared backbone model (PEGASUS) on the filtered datasets. ROUGE score and Entity(Summ) are mainly used to measure the word-level saliency and entity-level saliency, respectively. Entity(Doc) is used to measure the entity-level factual consistency. Red represents the lowest among all the three models, while Green represents the highest.

Entity-level factual consistency metric: measures the named entity matching between the generated summaries and the source documents. (Nan et al., 2021) With $NE(D)$ and $NE(S_{gen})$ representing the named entities in the source document and generated summaries, respectively, *Source-precision*(src_p) measures how many entities in the generated summaries are from the source documents, i.e. $src_p = |NE(D) \cap NE(S_{gen})|/|NE(S_{gen})|$. It is an evaluation metric for entity-level factual consistency, as it directly measures how consistent the generated summaries are with the source.

4.3 Datasets

We test and compare our SpanCopy model with the original PEGASUS on four datasets, in the domains of news (CNNDM(Nallapati et al., 2016), XSum(Narayan et al., 2018)) and scientific papers (Pubmed and arXiv(Cohan et al., 2018)). As a sanity check, we initially assess our models on subsets of these datasets, where all the entities in the reference summaries belong to the source document (we call these filtered datasets). In these cases ($src_p(gt) = 1$), Spam Copy and GR should dominate PEGASUS, because by design they tend to generate entities from the source document. We compare the size of filtered and original datsets in Table 3.

The statistics of the filtered and original datasets, on the lengths and number of entities in the document and summaries, can be found in Table 2. $src_p(gt)$ measures the entity-level factual consistency between the source document and the ground-truth summary, with lower value meaning that there are more novel entities in the ground-truth summaries. The table shows that the datasets in the news domain have higher density of the entities with respect to the lengths (number of words) of both documents and ground-truth summaries, i.e. N_{doc}/L_{doc} and N_{summ}/L_{summ} are larger for the news articles. a possible explanation is that news articles tend to describe an event or a story, which may contain more names of people, organizations, locations, etc., as well as dates. Interestingly, CNNDM and Pubmed contain less novel than the other two datasets (with higher $src_p(gt)$), something that the proposed SpanCopy mechanism may benefit from. Comparing the filtered datasets with the original ones, we can see that the number of entities in the summaries drops for all the datasets, especially for arXiv, as the more entities in the summary, the less likely they can be all matched to the source documents.

4.4 Results and Analysis

The results on the filtered and original datasets are shown in Table 4 and Table 5.

Model	ROUGE			Entity(Summ)			Entity(Doc)
	R-1	R-2	R-L	sum_r	sum_p	sum_f	src_p
CNNDM							
PEGASUS	44.62	20.82	31.05	46.87	42.25	42.29	89.92
SpanCopy	44.19	20.86	31.19	43.15	43.87	41.25	91.89
SpanCopy+GR	44.16	20.61	30.97	42.72	43.34	40.79	91.31
XSum							
PEGASUS	46.65	23.47	38.67	41.09	44.43	40.96	41.23
SpanCopy	46.23	22.76	37.96	39.90	42.97	39.70	41.89
SpanCopy+GR	46.02	22.36	37.58	40.12	42.66	39.67	42.79
Pubmed							
PEGASUS	46.11	19.43	41.22	22.12	24.81	20.61	67.03
SpanCopy	46.21	19.86	41.51	23.47	25.10	21.29	68.91
SpanCopy+GR	46.27	19.82	41.59	23.34	25.29	21.39	66.91
arXiv							
PEGASUS	44.23	16.55	39.15	20.98	25.42	20.56	52.70
SpanCopy	44.05	16.76	38.91	20.65	25.46	20.39	56.88
SpanCopy+GR	44.00	16.87	38.92	20.01	25.75	20.15	54.21

Table 5: Result of our models and the compared backbone model (PEGASUS) on the unfiltered datasets. See Table 4 for the details of the columns.

Model	R_{avg}	sum_f	src_p
CNNDM			
SpanCopy	-0.08	-1.04	+1.97
SpanCopy+GR	-0.25	-1.50	+1.39
XSum			
SpanCopy	-0.61	-1.26	+0.66
SpanCopy+GR	-0.94	-1.29	+2.16
Pubmed			
SpanCopy	+0.27	+0.68	+1.88
SpanCopy+GR	+0.31	+0.78	-0.12
arXiv			
SpanCopy	+0.20	+1.47	+3.99
SpanCopy+GR	+0.30	-0.72	-0.57
Overall (avg. across all datasets)			
SpanCopy	-0.06	-0.04	+2.13
SpanCopy+GR	-0.15	-0.68	+0.72

Table 6: The relative ROUGE score (avg of R-1, R-2 and R-L), the entity-level summary-f1 and source-precision of our models, compared with the PEGASUS model on the four datasets (original). The last block shows the overall performance for all the datasets.

Filtered Datasets We first evaluate our models, with the backbone model, PEGASUS on the filtered datasets, which is an easier task, and the results can be found in Table 4. All the models are fine-tuned and tested on the filtered datasets. Since we only keep the examples with all the entities in the summaries being matched with the entities in the source documents, the theoretical ceiling of src_p is 100. Comparing SpanCopy and PEGASUS, SpanCopy performs better than PEGASUS regarding both saliency and entity-level factual consistency. Plausibly, this is because all the entities in the ground-truth summary can be copied from the source document, in which case the SpanCopy mechanism can better learn to copy. The SpanCopy model with the GR component performs better regarding the entity-level saliency on three out of all the four datasets. On arXiv, the performance of SpanCopy with the GR component regarding both entity-level saliency and factual consistency is quite low. One likely reason might be that it is a rather difficult task to identify the salient entities in the arxiv dataset, as there is a large amount of entities in the source documents, but only very few entities are summary-worthy (164.1 v.s. 2.3 as shown in Table 2), which might bring in excessive noise.

Original Datasets In a second set of experiments, we fine-tune and test on the full/original datasets. On this realistic and more challenging task results are encouraging. As shown in Table 5, when the SpanCopy model is compared to PEGASUS, it improves the factual consistency of generated summaries with the source documents (src_p) on all the datasets, maintaining a very similar performance on the saliency metrics, i.e. ROUGE and entity-level saliency. Comparing across the four datasets, SpanCopy outperforms PEGASUS on both the saliency and factual consistency metrics on the Pubmed dataset. For better comparison, we show the rel-

Entities in the Source Document: Yemen(0.28), Americans(0.25), Saudi Arabia(0.23), the State Department(0.23), CNN(0.20),..., U.S.(0.15), ...
Ground-truth Summary: No official way out for Americans stranded amid fighting in Yemen. U.S. Deputy Chief of Mission says situation is very dangerous so no mass evacuation is planned .
PEGASUS: CNN's Ivan Watson joins a mother and her grandchildren waiting to be evacuated from Yemen. The State Department has said it is too risky to evacuate Americans from the area. Watson meets Americans who were on a CNN ship that docked at a Yemeni port.
SpanCopy: Dozens of Americans are trapped in Yemen. The U.S. has said it is too dangerous to evacuate Americans.
SpanCopy+GR: The U.S. has said it is too dangerous to evacuate Americans from Yemen. The State Department said it is too risky to conduct an evacuation of citizens. A group of U.S. organizations have filed a lawsuit against the government's stance on evacuations.

Table 7: Example of the entity-level factual inconsistency, taken from the CNNDM dataset. The first block shows the entities in the source document with high GR scores (shown in parenthesis) from the SpanCopy + GR model.

ative gains/loss regarding PEGASUS on all the datasets, as well as the overall average results in Table 6. It is clear that the SpanCopy model performs much better regarding entity-level factual consistency (+2.13) with essentially no change in saliency (-0.06 on average ROUGE and -0.04 on entity-level saliency). Admittedly, despite the success of the GR component on the filtered datasets on both word-level and entity-level saliency, it fails to deliver any gain on the original datasets. A plausible explanation is that GR makes the model focus excessively on the entities in the source document, therefore penalizing generation of new, potentially summary-worthy, entities.

Comparing the entity-level factual consistency on the filtered datasets and the original datasets, the filtered datasets always have higher src_p than the original ones, and the gain is especially larger on the XSum and arXiv datasets, as both of them contain more entity-level hallucinations in the original datasets. Remarkably, the performance gain of the SpanCopy model over PEGASUS on the filtered XSum dataset is much larger on the original XSum datasets (7.98 v.s. 0.66) , which might be because original XSum is more abstractive, the entity-level guidance is especially helpful for the abstractive examples with consistent entities in the summary.

4.5 Qualitative Analysis

For illustration, we examine a real example from the CNNDM dataset in Table 7, which is a news article on the evacuation of Americans during the time of the crossfire of warring parties in Yemen. While all of the three system generated summaries are able to capture the main statement that 'it's too dangerous to evacuate the Americans', the person 'Ivan Watson' mentioned by PEGASUS's summary does not exist in the source document, i.e., it is an 'hallucinated' entity. Most likely, PEGASUS is generating such hallucination because 'Ivan Watson' is a senior CNN correspondent several time associated with Yemen in other news article in the training set, and the model automatically 'picked the entity from the memory' to generate the summary without tightly adhering to the given document. In contrast, both of our models do not contain entities that are not in the source document, as the SpanCopy mechanism tend to guide the model to use more the entities in the source document. In addition, with the GR component, although the generated summary contains more matched entities with the source document, it pushes the model too far towards copying entities which are not salient (e.g. *The State Department*).

5 Conclusion and Future Work

In this paper, we tackle the problem of entity-level factual consistency for abstractive summarization, by guiding the model to directly copy the summary-worthy entities from the source document through the novel SpanCopy mechanism (with the optional GR component), which can be integrated into any transformer-based generative frameworks. By running a sanity check on arguably easier subsets of four diverse summarization datasets, SpanCopy with GR is confirmed to perform better on both entity-level factual consistency and saliency. More tellingly, the experiments on the original test sets show that the SpanCopy mechanism can effectively improve the entity-level factual consistency with essentially no change in the word-level and token-level saliency. In the future, we plan to extend our approach towards controllable generation with given entities. Specifically, instead of using the learnt GR scores, the model could generate summaries with desired entities provided by human.

Limitation

In our method, we employ an existing NER tool (Spacy) to label the entities in both the source documents and the summaries, and the performance of the NER tool may have an influence on the results of the model. Thus a good in-domain NER tool may be required when the work is extended to some specific domains, e.g. medical text.

In addition, we use PEGASUS(Zhang et al., 2020) as our base model in all the experiments on different datasets, as it has delivered top performance on multiple summarization datasets. We follow the original paper on the length limits of all the datasets, however, the length of the source documents in both scientific paper datasets are much longer than the length limit (3k/6k v.s. 1024), which leaves the room for further improvement with sparse attention techniques applied (Xiao et al., 2021; Guo et al., 2022).

Ethics Consideration

Although we tackle the problem of factual inconsistency for abstractive summarization, and improve the entity-level factual consistency of the generated summaries by applying the entity-level span copy mechanism, the generated summaries still contain unfactual information. Therefore, caution must be exercised when the model is deployed in practical settings.

References

Bin Bi, Chenliang Li, Chen Wu, Ming Yan, Wei Wang, Songfang Huang, Fei Huang, and Luo Si. 2020. PALM: Pre-training an autoencoding&autoregressive language model for context-conditioned generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8681–8691, Online. Association for Computational Linguistics.

Meng Cao, Yue Dong, and Jackie Chi Kit Cheung. 2021. Inspecting the factuality of hallucinated entities in abstractive summarization. *CoRR*, abs/2109.09784.

Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2018. Faithful to the original: Fact aware neural abstractive summarization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

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

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

Esin Durmus, He He, and Mona Diab. 2020. FEQA: A question answering evaluation framework for faithfulness assessment in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5055–5070, Online. Association for Computational Linguistics.

Mandy Guo, Joshua Ainslie, David Uthus, Santiago Ontanon, Jianmo Ni, Yun-Hsuan Sung, and Yinfei Yang. 2022. LongT5: Efficient text-to-text transformer for long sequences. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 724–736, Seattle, United States. Association for Computational Linguistics.

Daniel King, Zejiang Shen, Nishant Subramani, Daniel S. Weld, Iz Beltagy, and Doug Downey. 2022. Don't say what you don't know: Improving the consistency of abstractive summarization by constraining beam search.

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, Online. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence 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.

Haoran Li, Song Xu, Peng Yuan, Yujia Wang, Youzheng Wu, Xiaodong He, and Bowen Zhou. 2021. Learn to copy from the copying history: Correlational copy network for abstractive summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4091–4101, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summariza-*

tion Branches Out, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

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

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

Yang Liu and Mirella Lapata. 2019b. 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 3730–3740, Hong Kong, China. Association for Computational Linguistics.

Yi Liu, Guoan Zhang, Puning Yu, Jianlin Su, and Shengfeng Pan. 2021. BioCopy: A plug-and-play span copy mechanism in Seq2Seq models. In *Proceedings of the Second Workshop on Simple and Efficient Natural Language Processing*, pages 53–57, Virtual. Association for Computational Linguistics.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

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.

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

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.

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, Vancouver, Canada. Association for Computational Linguistics.

Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2017. Abstractive document summarization with a graph-based attentional neural model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1171–1181, Vancouver, Canada. Association for Computational Linguistics.

Wen Xiao, Iz Beltagy, Giuseppe Carenini, and Arman Cohan. 2021. PRIMER: pyramid-based masked sentence pre-training for multi-document summarization. *CoRR*, abs/2110.08499.

Wen Xiao and Giuseppe Carenini. 2019. Extractive summarization of long documents by combining global and local context. 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 3011–3021, Hong Kong, China. Association for Computational Linguistics.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

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

Qingyu Zhou, Nan Yang, Furu Wei, and Ming Zhou. 2018. Sequential copying networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

A Model and Training Details

We use PEGASUS as our backbone model, which contains 571M parameters, and the span copy mechanism has 2M additional parameters. We train the fine-tuned models from the huggingface model hub[5] for 100k steps (16 data per step) with early stopping based on the ROUGE scores on the validation set, which takes around 24 hours with single V100 GPU.

[5] https://huggingface.co/models

B Software and Licenses

Our code is licensed under Apache License 2.0.
Our framework dependencies are:

- HuggingFace Datasets[6], Apache 2.0
- NLTK [7], Apache 2.0
- Numpy[8], BSD 3-Clause "New" or "Revised"
- Spacy[9], MIT
- Transformers[10], Apache 2.0
- Pytorch[11], Misc
- Pytorch Lightning [12],Apache 2.0
- ROUGE [13], Apache 2.0

[6] https://github.com/huggingface/datasets/blob/master/LICENSE

[7] https://github.com/nltk/nltk

[8] https://github.com/numpy/numpy/blob/main/LICENSE.txt

[9] https://github.com/explosion/spaCy/blob/master/LICENSE

[10] https://github.com/huggingface/transformers/blob/master/LICENSE

[11] https://github.com/pytorch/pytorch/blob/master/LICENSE

[12] https://github.com/PyTorchLightning/pytorch-lightning/blob/master/LICENSE

[13] https://github.com/google-research/google-research/tree/master/rouge