Hallucinated but Factual! Inspecting the Factuality of Hallucinations in Abstractive Summarization

Anonymous ACL submission

Abstract

State-of-the-art abstractive summarization systems often generate *hallucinations*; i.e., content that is not directly inferable from the source text. Despite being assumed incorrect, we find that much hallucinated content is factual, namely consistent with world knowledge. These factual hallucinations can be beneficial in a summary by providing useful background information. In this work, we propose a novel detection approach that separates factual from non-factual hallucinations of entities. Our method utilizes an entity's prior and posterior probabilities according to pre-trained and fine-tuned masked language models, respectively. Empirical results suggest that our approach vastly outperforms five baselines and strongly correlates with human judgments. Furthermore, we show that our detector, when used as a reward signal in an off-line reinforcement learning (RL) algorithm, significantly improves the factuality of summaries while maintaining the level of abstractiveness. [1]

1 Introduction

State-of-the-art abstractive summarization systems can generate fluent summaries with high automatic evaluation scores in terms of ROUGE (Lin, 2004). However, recent studies have shown that these systems are prone to hallucinate content that is not supported by the source document (Maynez et al., 2020; Kang and Hashimoto, 2020; Durmus et al., 2020; Zhao et al., 2020; Filippova, 2020; Kryscinski et al., 2020). For instance, Maynez et al. (2020) discovered that 64.1% of the summaries generated by a BERT-based abstractive summarization model on XSUM (Narayan et al., 2018a) contain hallucinations.

Previous studies commonly assume that hallucination is an undesirable behavior in abstractive summarization systems. They investigate the cause of model hallucination (Kang and Hashimoto, 2020; Wang and Sennrich, 2020) and propose methods that reduce the frequency of all hallucinations (Filippova, 2020; Zhao et al., 2020; Nan et al., 2021; Narayan et al., 2021).

Source: Under the proposals, 120,000 additional asylum seekers will be distributed among EU nations, with binding quotas. (...) **Mr Juncker** told the European Parliament it was "not a time to take fright". (...) He said tackling the crisis was "a matter of humanity and human dignity". "It is true that Europe cannot house all the misery in the world. But we have to put it into perspective." (...)
Generation: European Commission President Jean-Claude Juncker has set out his proposals for dealing with the migrant crisis in a speech to MEPs, saying Europe cannot house all the misery in the world.

Table 1: Example of factual hallucinations in a BART generated summary on XSUM. Neither the title "European Commission President" nor the first name "Jean-Claude" is mentioned in the document but both are factual.

Our stance in this paper is that *hallucinations are not always undesirable*: many factual hallucinations provide additional world knowledge that is important for summary comprehension. Table 1 presents one such example from XSUM: the hallucinated content *European Commission President* provides additional background information on the role of *Mr. Juncker*. Factual hallucinations refer to content that is verifiable by world knowledge but not inferable from source text.

We thus argue that not all hallucinations should be treated equally; in particular, factual hallucinations may be less deleterious or even potentially beneficial to to be included in a summary, as opposed to non-factual ones. We propose a method to classify entities according to whether they are hallucinations and whether they are factual (if hallucinated). We focus on entities (e.g., persons, locations, dates, cardinal numbers) because they are necessary to express the most salient pieces of information in a summary. Moreover, entity halluci-

[1] Both the data and code will be made publicly available after the anonymity period.

nations are common in generated summaries. As we will show later in our work, about 30% of entities generated by BART (Lewis et al., 2020) on XSUM test set are hallucinated.

Our approach is inspired by the observation that many hallucinated entities are generated with low probabilities. This observation suggests that the summarization model's confidence correlates with the factuality statuses of generated entities. In other words, the uncertainty is indicative of the likelihood of whether generated entities are hallucinated and non-factual.

We refer to the probability of an entity being in a summary without considering the source document as its prior probability, and its probability given the document as its posterior probability. Our assumption is that if an entity in a generated summary results in a factual error, giving the source should not provide more evidence for it, resulting in a small change in probability between the prior and the posterior. Based on this assumption, we propose to use the prior and posterior probabilities as the key features in a simple classifier that predicts an entity's hallucination status and factuality.

Due to the lack of fine-grained hallucination annotation, we create an entity-level hallucination and factuality annotation on the XSUM dataset. We evaluate our detection method on this annotated dataset as well as annotations from Maynez et al. (2020). On both datasets, our approach outperforms five baseline models at identifying non-factual hallucinations. In addition, our approach has a strong correlation with the factuality scores given by human judges. Besides, we show that our detector, when used as a reward signal in training neural-based summarizers with the off-line RL algorithm, significantly improves the factuality of generated summaries even when the underlying dataset is noisy.

Our contributions are the following: (*i*) We demonstrate that an entity's prior and posterior probabilities can be used to infer whether it is hallucinated and factual. Based on this hypothesis, we propose a novel approach for entity-level hallucination detection and factuality checking. Our approach outperforms five baselines from previous work on two human-annotated datasets, in addition to having a strong correlation with summary-level factuality scores given by human judges. (*ii*) We empirically demonstrate that our classifier can provide reliable reward signals for RL algorithms, leading to improved factuality while maintaining the level of abstractiveness in generated summaries. (*iii*) We create a set of entity-level hallucination annotations.

2 Related Work

The correctness of summarization systems' outputs has been evaluated as one aspect of content selection in the past, for example using the Pyramid method (Nenkova and Passonneau, 2004). As neural abstractive summarizers have become popular, their issues with correctness have sparked much recent work that focus specifically on model hallucinations and summary factuality (Kryscinski et al., 2020).

2.1 Model Hallucination

Maynez et al. (2020) conducted a large-scale human evaluation of several neural abstractive summarization systems, and found that hallucinations are common among the outputs of different summarization models.

Recently, many methods have been proposed to reduce model hallucination. Kang and Hashimoto (2020) propose a "loss truncation" training algorithm that filters out noisy training samples which may lead a model to hallucinate. Zhao et al. (2020) use a verification system to recognize non-factual quantities in summaries and adopt a re-ranking system to reduce the number of hallucinated quantities in the final output summary. Narayan et al. (2021) use entity chains to mitigate the hallucination problem in the generation of abstractive summaries. Nan et al. (2021) show that data filtering and use a summary-worthy entity classification task as an auxiliary training objective can help improve model's entity-level factuality.

Filippova (2020) proposed a method for controlling hallucination in data-to-text generation task. They suggest that a conditional language model (CLM) will put more probability mass on a non-hallucinated entity than an unconditional language model (LM). Our work differs in that we focus on both hallucination and factuality. Also, our method works at the entity-level rather than the sentence-level, and is geared towards text summarization.

2.2 Summary Factuality

Another line of work focuses on evaluating the factual consistency of abstractive summarization systems. Kryscinski et al. (2020) train models on an artificially corrupted dataset for factual errors detection. Cao et al. (2020) induce artificial perturbations in text to train a summary error correction system, but find that there is a large gap between such artificial perturbations and the type of hallucinations that are generated by abstractive summarizers. (Goyal and Durrett, 2020) measure factual consistency by checking whether the semantic relationship manifested by individual dependency arcs in the generated summary is supported by the source document. Wang et al. (2020); Dong et al. (2020); Durmus et al. (2020) measure and improve the factual consistency of summaries by asking and answering questions based on generated summaries and input documents.

3 Method

In this section, we propose a novel detection approach that separates factual from non-factual hallucinations of entities (Section 3.2), and present a factuality-aware training framework for summarization models trained on noisy dataset (Section 3.3).

3.1 Problem Statement

Let (S, R) be a pair of a source document and a reference summary, where $S = (s_1, ..., s_M)$ is the source document with M tokens, and $R = (r_1, ..., r_L)$ is the reference summary with L tokens. Let $G = (g_1, ..., g_N)$ be the model-generated summary with N tokens. For each named entity e_k, which we assume to be a span of tokens $g_{i_k}, ..., g_{i_k+|e_k|-1}$ ($|e_k| \geq 1$) starting at position i_k in G, the task is to determine whether e_k is hallucinated, and whether it is factual. We define an entity as hallucinated if it is not directly inferable in its generated context given the input document S. If an entity is hallucinated, we further classify it into two subtypes: *factual hallucinations* and *non-factual hallucinations*. Factual hallucinations cannot be directly entailed in their generated context from the source document but can be based on world knowledge (see Table 1). Non-factual hallucinations are entities that are neither inferable from the source nor based on world knowledge.

3.2 The Prior & Posterior Probability of an Entity

We now define the prior and posterior probabilities of an entity, which we will use to predict its hallucination and factuality statuses.

For entity e_k, we define its prior probability $p_{\text{prior}}(e_k)$ as the probability of its generation by a language model that does not have access to the source text. If e_k spans multiple tokens, we compute its probability auto-regressively. Let c_k be the context of entity e_k in G, excluding the tokens in e_k. Then:

$$\begin{aligned} p_{\text{prior}}(e_k) &= fP_{\text{MLM}}(e_k \mid c_k) && (1) \\ &= \prod_{t=1}^{|e_k|} P_{\text{MLM}}(e_k^t \mid e_k^{1...t-1}, c_k) && (2) \end{aligned}$$

which we compute using a masked language model P_{MLM}.

The posterior probability $p_{\text{pos}}(e_k)$ of entity e_k is the conditional probability of the entity given the context and the source text:

$$\begin{aligned} p_{\text{pos}}(e_k) &= P_{\text{CMLM}}(e_k \mid c_k, S) && (3) \\ &= \prod_{t=1}^{|e_k|} P_{\text{CMLM}}(e_k^t \mid e_k^{1...t-1}, c_k, S), && (4) \end{aligned}$$

where CMLM is a conditional masked language model. CMLM is an encoder-decoder model that is trained with a masked language model objective on a parallel dataset. Specifically, a CMLM predicts a target sequence T given a source text S and part of the target T_{masked}, where T_{masked} is the target sequence with a random entity being masked. In order to correctly generate the missing part of the sentence, the model needs to condition on both T_{masked} and S. Alternatively, we can calculate the entity's posterior probability using a conditional language model (CLM) instead of a CMLM. In this case, the entity's posterior probability is defined as $P_{\text{CLM}}(e_k \mid c_{e_k}, S)$ where $c_{e_k} = g_1, ..., g_{i-1}$. Note that CLM is only conditioned on the left context.

Training a Discriminator To classify the hallucination and factuality statuses of a given entity, we need to train a discriminator model. We use the K-Nearest Neighbors (KNN) algorithm since it requires no training and makes minimal assumptions about the form of the decision boundary, as a non-parametric method. It also offers adequate interpretability. The KNN classifier is trained using the prior and posterior probabilities as features on our labeled dataset. Since the classifier is used for entity hallucination and factuality assessment, we refer to it as **EntFA**. Besides using the prior/posterior probability as features, we also add a binary overlap feature that indicates whether the entity appears in the document. We train two classifiers for hallucination detection and factuality checking tasks respectively.

3.3 Improving the Factuality of Abstractive Summarization Systems

We now propose a factuality-aware training approach for summarization systems that combines our factuality assessment model with the latest offline RL technique.

RL for Text Generation Sequence generation of the tokens in the summary text can be viewed as a finite Markov Decision Process (MDP). At each time-step t, the state s_t consists of the source text x and the previously generated tokens $y_{<t}$, $s_t = (y_{<t}, x)$. The agent, which is the summarization model, takes an action by generating a new token a_t. Depending on the action taken, the agent gets a reward $r_t = R(s_t, a_t)$ and deterministically transitions to the next state $s_{t+1} = (y_{<t+1}, x)$. The probability of each action (i.e., token) is specified by the policy $\pi_\theta(a_t|s_t)$. The goal of the agent is to maximize the discounted cumulative reward throughout the trajectory: $J(\theta) = \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{T} \gamma^t r_t\right]$.

When training the summarization model with human-written reference summaries, we can frame the training process as an off-line RL problem with expert demonstrations (i.e., the reference summaries). In this setting, since we are sampling trajectories from a behavior policy, we need an importance sampling term w_t to correct the gradient estimation. Following Pang and He (2021)'s work, we approximate w_t with $\pi_\theta(a_t|s_t)$ and this gives us the following objective:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau\sim\pi_b}\left[\sum_{t=0} \pi_\theta(a_t|s_t) \nabla_\theta \log \pi_\theta(a_t \mid s_t) \hat{Q}(a_t, s_t)\right] \tag{5}$$

where $\hat{Q}(a_t, s_t) = \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'}$ is the estimated return from state s_t and π_b is the behavior policy from which we draw trajectories τ. In our case, π_b is the (noisy) summarization dataset.

Training with a Factuality-based Reward One problem in the off-line RL setting is that expert demonstrations, which in our case are the reference summaries, are often noisy and contain content that cannot be inferred from the source document. The commonly used teacher forcing training encourages the model to blindly imitate the training data, which leads to model hallucination at inference time (Kang and Hashimoto, 2020).

To discourage the model from overfitting to the noise in the training set, we use the predictions from our classifier as factuality reward signals to guide the training of the summarization model. In the off-policy learning stage, we use our factuality classifier to label all the entities in the training set. If an entity is classified by our classifier as "non-factual", we consider it noise and give it a negative reward $-r_{\text{nfe}}$. For factual entities and other tokens, we use the posterior probability from a MLE-trained model as token-level rewards, as in (Pang and He, 2021). Formally, we have:

$$R(s_t, a_t) = \begin{cases} -r_{\text{nfe}}, & \text{if } a_t \text{ is non-factual} \\ p_{\text{MLE}}(a_t|s_t), & \text{otherwise} \end{cases}$$

4 Dataset

4.1 XEnt dataset

To study entity hallucination and factuality in abstractive summarization, we need annotations of entity- or token-level hallucination. To the best of our knowledge, there is no such dataset available. Therefore, we create a dataset ourselves, which we call the XEnt dataset.

We[2] annotate 800 summaries generated by BART, which is one of the state-of-the-art abstractive summarization models. The input documents are randomly selected from XSum test set. We choose XSum because it is more abstractive than other summarization datasets. We extract 2,838 entities from the 800 generated summaries. We randomly select 30% of the samples as our test set.

We manually labeled each entity with one of the following three tags: non-hallucinated, factual hallucination, and non-factual hallucination. First, we extract entities from the given summary using automatic NER tools (Honnibal and Montani, 2017). Then, we check whether each property associated with the identified entity can be directly entailed using the information from the source document. If so, then the property is non-hallucinated. For instance, consider the entity "European Commission President Jean-Claude Juncker" in Table 1. The last name "Juncker" can be directly entailed from

[2] Two coauthors and three graduate students. The data collection process was approved by institution ethics committee.

Category	Source Document	Generated Summary
Non-hallucinated	(...) Tian Tian has had cubs in the past in China, before she came on loan to Edinburgh. (...) The panda enclosure at Edinburgh Zoo is due to close to visitors from Saturday ahead of a possible birth.	Edinburgh Zoo's giant panda, Tian Tian, could give birth at the end of the month.
Factual Hallucination	The couple, who have been dating since 2011, wed in front of about 10 people in Mazan, Provence - close to where the bride's family has a holiday home. (...) Knightley, 28, announced her engagement to Righton, 29, last year. "Keira was a charming bride, very modest and simple in her attitude, as was James," (...)	Oscar-winning actress Keira Knightley and British musician James Righton have married in a small ceremony in France.
Non-factual Hallucination	The city was brought to a standstill on 15 December last year when a gunman held 18 hostages for 17 hours. Family members of victims Tori Johnson and Katrina Dawson were in attendance. (...) Prime Minister Malcolm Turnbull gave an address saying a "whole nation resolved to answer hatred with love". (...)	Sydney has marked the first anniversary of the siege at the Waverley cafe in which two women were killed by a gunman in the Australian city.
Intrinsic Hallucination	Christopher Huxtable, 34, from Swansea, had been missing since the collapse in February. His body was found on Wednesday and workers who carried out the search formed a guard of honour as it was driven from the site in the early hours of the morning. (...)	The body of a man whose body was found at the site of the Swansea Bay Power Station collapse has been removed from the site.

Table 2: Examples of four types of hallucinations. In the second example, the nationality of the groom and the country where the wedding took place are not directly stated in the source. According to information online both entities are factual. In the third example, the terrorist attack described in the news took place at a place called "Lindt Cafe" according to Wikipedia. Therefore, "the Waverley cafe" in the generated summary is non-factual.

the source document. Therefore, it is not a hallucination. However, the first name "Jean-Claude" and the position information "European Commission President" are not mentioned in the source. In the next step, we need to decide whether these information is factual or not using world knowledge. This often requires external resources such as Wikipedia or Google Search. In this case, "European Commission President" and "Jean-Claude" are both factual. If there is no information found online to prove or disprove the hallucinated entity, it is labeled as non-factual. There is a special case where the entity misrepresents information from the document. For instance, the summary might include a number from the document but that number is actually related to a different event. In this case, the entity is considered as an intrinsic hallucination (Maynez et al., 2020). In this work, we will focus on extrinsic hallucinations, so we discarded all intrinsic hallucinations in our experiments. Table 3 shows the distribution of entities by hallucination and factuality status in our labeled dataset. We show an example for each hallucination type in Table 2.

Inter-Annotator Agreement We report Fleiss's Kappa (κ) to access the reliability of agreement between annotators. Each sample in the dataset is annotated by three different annotators. We obtain a high agreement ($0.80 \leq \kappa \leq 1.00$) with $\kappa = 0.809$. Following Pagnoni et al. (2021), we also report the percentage μ of annotators that agree with the majority class. We obtain $\mu = 0.931$ of annotators agreeing with the majority class on the four-category annotation which shows substantial agreement.

4.2 MENT Dataset

Recently, Maynez et al. (2020) released a set of factuality and hallucination annotations for XSUM. For each generated summary, they labeled the hallucinated spans as well as the overall factuality of the summary. Compared with our labeling approach, their annotation has a lower granularity and does not distinguish between factual and non-factual hallucination. Therefore, we have to convert their dataset first before using it for evaluation.

To perform entity-level factuality checking on their dataset, we do the following: First, we extract entities from the annotated summaries. For entities that are extracted from factual summaries, we label them as factual entities. For each entity from non-factual summary, if it is inside an extrinsic hallucinated span, then we assume the entity is non-factual. Otherwise the entity is labeled as a factual. This process gives us a new dataset that has the same format as ours for entity-level factuality evaluation. We refer to this new dataset as the MENT dataset.

However, it is worth pointing out that the converted dataset is noisy. For instance, in Maynez et al. (2020)'s annotation, the entire generated summary is often labeled as a hallucinated span if it does not capture the meaning of the document well. In this case, the hallucinated span could still contain faithful entities with respect to the source document. This could result in false-positive non-factual entities after the conversion. Therefore, we filter out entities in the extrinsic hallucination span that also appear in the source document.

5 Evaluation Tasks

5.1 Entity-level Hallucination & Factuality Classification

We evaluate our method on entity-level hallucination and factuality classification tasks on XENT and MENT. For each entity in the summary, the model predicts a hallucination label and a factuality label. We will conduct factual and hallucination assessments separately for comparison with the baselines. We compare our method with five baselines models, which are discussed in detail in Section 6.1.

5.2 Correlation with Human Judgments of Factuality

In addition to entity-level classification performance, we also evaluate our methods by correlating them against human judgments of factuality. Previous work has collected summary-level judgments of factuality from human annotators, which are then correlated with automatic evaluation measures applied to those summaries. To apply our entity-level method, we use the lowest classifier confidence for the factual class among its entities as the factuality score for the entire summary. We evaluate correlation on two datasets by Pagnoni et al. (2021) and Wang et al. (2020).

5.3 Evaluating the Factuality of Summarization Systems

To evaluate our factuality-aware training approach proposed in Section 3.3, we train a summarization model with factuality rewards and evaluate model's predictions on XSUM test set. To evaluate the faithfulness of generated summaries, we use automatic faithfulness evaluation tools FEQA (Durmus et al., 2020) and DAE (Goyal and Durrett, 2020)[3]. We also calculate ROUGE scores, and the percentage of n-grams and percentage of entities in the generated summaries that are not found in the source document (ENFS). The percentage of novel n-grams reflects the extractiveness of summarization model.

Label	#Samples	Total Ent.
Non-hallucinated	1,921 (67.69%)	2,838
Factual hal.	441 (15.54%)	
Non-factual hal.	421 (14.83%)	
Intrinsic hal.	55 (1.94%)	

Table 3: Statistics of labeled dataset. See Appendix A.2 for more details.

	Hallucination		Factuality	
	Acc.	F1	Acc.	F1
Overlap-based	92.93	91.73	81.25	74.19
Synonym-based	90.76	89.42	81.30	74.79
Alignment	78.35	71.10	81.65	66.03
LM-based	74.18	54.99	84.54	57.80
Zhou et al. (2020)	86.66	81.71	85.76	75.07
ENTFA (ours)	**93.09**	**91.91**	**90.95**	**81.82**

Table 4: Entity's factuality and hallucination status evaluation results on XENT. We report the accuracy and (macro) F1 score on the test set. The number of neighbors k is set to 20 for both tasks.

6 Experiments

Training CMLM & MLM For training the CMLM, we use both XSUM, Narayan et al. (2018b)) and the CNN/Dailymail dataset (Hermann et al., 2015) dataset. To build a training corpus for CMLM, we randomly select one entity in each reference summary and mask it with a special [MASK] token. We append a [S] token at the beginning of each summary. The document and summary are concatenated together (separated by [\S] token) as CMLM's input. The training target is the reference summary without any masking. If there is no specification, we use the CMLM trained on XSUM. For the MLM, we use the large BART model. BART is pre-trained on five different reconstruction tasks including token masking and text infilling. For more experimental setup and hyper-parameter setting details, see Appendix A.3.

6.1 Classification Experiments

Baselines We compare with four baseline methods: **(1)** The *overlap-based* method checks the word overlap between the summary and the source document. In our case, we check whether a given entity in the generated summary also exist in the source document. If it does not, the entity is classified as both hallucinated and non-factual. **(2)** The *synonym-based* baseline extends the *overlap-based* baseline by checking the overlap of sum-

[3] In this work, we define the faithfulness of the summary as whether it is faithful with respect to the source. Factuality as whether is factual with respect to world knowledge.

mary synonyms and source synonyms. See Zhou et al. (2020) for more details. **(3)** The *alignment-based* baseline is based on the unsupervised word alignment method SimAlign by Jalili Sabet et al. (2020). SimAlign extracts word alignments from similarity matrices induced from pretrained embeddings. In our task, we treat all unaligned entities in summaries as hallucinated and non-factual. **(4)** The *LM-based* method is proposed by Filippova (2020). The *LM-based* method uses LM and CLM to compute the token's prior and posterior probability. In Filippova (2020)'s work, they compare the value of p_{prior} and p_{pos}. If the generated token does not match the reference and p_{prior} is greater than p_{pos}, the token is classified as hallucinated. Since we are evaluating the generated summary but not the reference, we modify their method to the following: if the entity is not found in the source and $p_{\text{prior}} > p_{\text{pos}}$, then the entity is classified as non-factual and hallucinated. **(5)** Zhou et al. (2020) frame the hallucination detection task as a sequence labeling task. They train a hallucination labeling model on synthetic data. We adapt their model to our task by finetuning their model on XENT.

Evaluation Results on XENT Table 4 shows the evaluation results of our classifiers and baselines in terms of both entity factuality and hallucination status classification. The results show that our approach outperforms five baselines on the factuality classification task. To show that our model is statistically better than the baselines, we run a 10-fold cross-validated paired t-test comparing our model with five baselines. The results show that our model is better than the baseline models with p-value less than $3.27e-5$. On the hallucination detection task, the overlap-based and synonym-based baselines achieve relatively high accuracy. However, these methods cannot distinguish between factual and non-factual hallucinations. This is the reason for their performance degradation on factuality classification task. For hallucination classification, the reason computing word overlap with the source does not completely solve the hallucination detection problem is that hallucination is defined based on the semantic relationship between the source and the summary. There can exist words that are not in the source document but which can nevertheless be inferred from it.

Evaluation Results on MENT Dataset Table 5 shows the evaluation results on MENT. ENTFA are learned on our annotated training set with k set to 20. The performance of all models is lower on this dataset. This may be due to fact that the converted dataset is noisier than the XENT dataset (see Section 4.2). For the factuality classification task, our model outperforms five baseline models. This demonstrates the generalizability of our approach.

	Acc.	**F1**
Overlap-based	68.22	54.68
Synonym-based	68.91	53.43
Alignment	69.21	50.86
LM-based	67.48	48.02
Zhou et al. (2020)	71.02	56.42
ENTFA (ours)	**78.48**	**60.23**

Table 5: Entity-level factuality evaluation results on converted MENT Dataset (Maynez et al. (2020)).

Metric	FRANK (Partial Pearson's ρ)	Wang et al. (PCC)
BLUE	0.139	0.118
ROUGE-1	0.155	0.132
ROUGE-L	0.156	0.089
METEOR	0.155	0.100
BERTScore	-0.0359	0.025
QAGS	-0.0225	0.175
FEQA	0.0242	-
DAE	0.0444	-
ENTFA (ours)	**0.183**	**0.268**

Table 6: Summary-level Pearson correlation coefficients between various automatic metrics and human judgments of factuality for XSUM datasets. In the middle column, we use the FRANK benchmark for factuality evaluation metrics from Pagnoni et al. (2021); In the right column, we use the human judgments collected by Wang et al. (2020). All baselines' coefficient values are cited from their papers.

6.2 Correlation Experiments

Table 6 presents the correlation evaluation results. On Pagnoni et al. (2021)'s benchmark dataset, our approach has the highest partial Pearson correlation coefficient $\rho = 0.183$ ($p < 1e^{-8}$). On Wang et al. (2020)'s dataset (right column), our approach outperforms all other automatic metrics significantly. These results indicate that our model can be used for automatic factuality evaluation of summaries at both the entity and sentence levels.

6.3 Factuality Evaluation Results of Summarization Systems

Baselines We compare our approach with four baselines: a teacher forcing trained summarizer (MLE), a RL-based summarizer (RL) (Pang and

System	ROUGE R1 ↑	 RL ↑	% of novel n-gram unigrams ↑	 bigrams ↑	Faithfulness % ENFS ↓	 FEQA ↑	 DAE ↑	ENTFA % Factual Ent ↑	 % Factual Hal ↑
MLE	45.1	37.3	27.86	74.47	42.0	25.9	34.6	82.8	21.4
RL	**45.8**	**37.6**	28.14	74.73	43.2	25.6	33.3	82.8	21.6
LM-based	43.2	34.6	**29.75**	**75.86**	38.2	24.2	31.3	87.4	21.7
Loss trunc (c=0.3)	44.1	36.0	26.82	73.39	41.3	26.3	36.4	83.9	21.3
Loss trunc (c=0.7)	42.7	34.8	26.61	73.19	40.6	26.7	38.8	84.1	20.7
Ours ($r_{\text{nfe}} = 2.0$)	44.6	36.2	27.71	74.90	37.2	26.5	37.3	90.1	**24.0**
Ours ($r_{\text{nfe}} = 4.0$)	43.0	34.9	26.87	74.11	**32.8**	**27.3**	**40.8**	**92.5**	22.4

Table 7: Comparison of different summarization models. Results are evaluated on XSUM's official test set. "% Factual Ent" and "% Factual Hal" are the percentage of factual entities and factual hallucinations classified by ENTFA model respectively. "% ENFS" is the percentage of entities in generated summary that not found in source document. For the loss truncation baseline, c is the percentage of data to be dropped.

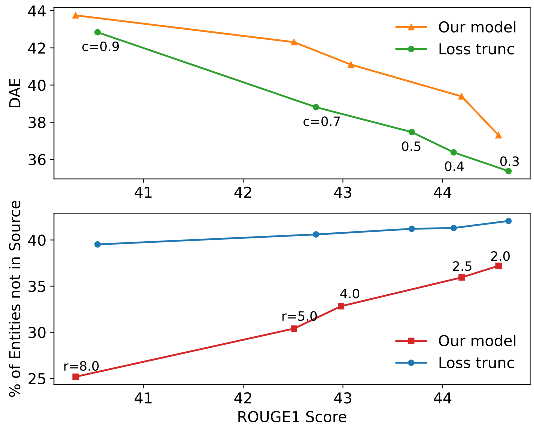


Figure 1: The factuality and ROUGE score trade-off curve on XSUM. We use different reward value r_{nfe} for our approach and different drop rate c for the loss truncation baseline.

He, 2021) and a summarizer trained with the loss truncation technique from Kang and Hashimoto (2020). We also replace our factuality assessment model ENTFA with Filippova (2020)'s approach (LM-based) for entity factuality labeling as another baseline model (see Section 3.3).

Table 7 shows the evaluation results on XSUM. The results show that our approach outperforms all baselines with fewer non-factual entities and higher faithfulness scores. Note that our approach has the lowest ENFS rate while having the highest percentage of factual hallucinations. Compared with the loss truncation baseline, our method also produces more novel n-grams. These show that our method does not improve the factuality of the model by simply making the model more extractive.

Figure 1 shows the factuality and abstractiveness trade-off curves of our model compared to the loss truncation baseline. At the same level of ROUGE performance, our method can obtain a higher factuality score. This further proves that our model can generate both factual and high-quality summaries compared with the loss truncation baseline.

	Factuality	**Hallucination**
ENTFA	81.82	91.91
w/o overlap	77.18	74.83
w/o prior	80.12	91.32
w/o posterior	70.30	91.12

Table 8: Ablation studies of different feature combination. We report the F1 score on XENT test set.

7 Ablation Studies

To explore the effect of each feature, we conduct an ablation study by training the KNN classifier with fewer features. The results are illustrated in Table 8 and show that all the proposed features are useful. For factuality classification, The performance w/o posterior drops significantly from 90.95 to 85.69. This result suggests that the posterior probability is crucial for factuality classification. For hallucination classification, the overlap-based feature has the most significant impact on model performance.

8 Conclusion

In this paper, we investigate the hallucination and factuality problems in abstractive summarization. We show that about 30% of entities generated by state-of-the-art summarization model are hallucinated. More interestingly, more than half of the hallucinated entities are factual with respect to the source document and world knowledge. We propose a novel method based on the entity's prior and posterior probabilities according to masked language models. Our approach outperforms five baseline models on both factuality classification and hallucination detection tasks on human-annotated datasets. In addition, using our classifier as a reward signal vastly improves the factuality of summarization systems. Our approach is limited to entity-level hallucination and factuality classification. In the future, we are interested in extending our work to arbitrary text spans.

References

Meng Cao, Yue Dong, Jiapeng Wu, and Jackie Chi Kit Cheung. 2020. Factual error correction for abstractive summarization models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6251–6258, Online. 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.

Katja Filippova. 2020. Controlled hallucinations: Learning to generate faithfully from noisy data. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 864–870, Online. Association for Computational Linguistics.

Tanya Goyal and Greg Durrett. 2020. Evaluating factuality in generation with dependency-level entailment. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3592–3603, Online. Association for Computational Linguistics.

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*, volume 28, pages 1693–1701. Curran Associates, Inc.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Masoud Jalili Sabet, Philipp Dufter, François Yvon, and Hinrich Schütze. 2020. SimAlign: High quality word alignments without parallel training data using static and contextualized embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1627–1643, Online. Association for Computational Linguistics.

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

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

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.

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

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.

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. 2018a. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018b. 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*, Brussels, Belgium.

Shashi Narayan, Yao Zhao, Joshua Maynez, Gonçalo Simoes, and Ryan McDonald. 2021. Planning with entity chains for abstractive summarization. *arXiv preprint arXiv:2104.07606*.

Ani Nenkova and Rebecca J. Passonneau. 2004. Evaluating content selection in summarization: The pyramid method. In *Proceedings of the Human Language*

Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004, pages 145–152.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. 2021. Understanding factuality in abstractive summarization with FRANK: A benchmark for factuality metrics. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, Mexico City.

Richard Yuanzhe Pang and He He. 2021. Text generation by learning from demonstrations. In *International Conference on Learning Representations*.

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.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition,*.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and answering questions to evaluate the factual consistency of summaries. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5008–5020, Online. Association for Computational Linguistics.

Chaojun Wang and Rico Sennrich. 2020. On exposure bias, hallucination and domain shift in neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3544–3552, Online. Association for Computational Linguistics.

Jiacheng Xu, Shrey Desai, and Greg Durrett. 2020. Understanding neural abstractive summarization models via uncertainty. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6275–6281, Online. Association for Computational Linguistics.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339. PMLR.

Zheng Zhao, Shay B. Cohen, and Bonnie Webber. 2020. Reducing quantity hallucinations in abstractive summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2237–2249, Online. Association for Computational Linguistics.

Chunting Zhou, Jiatao Gu, Mona Diab, Paco Guzman, Luke Zettlemoyer, and Marjan Ghazvininejad. 2020. Detecting hallucinated content in conditional neural sequence generation. *arXiv preprint arXiv:2011.02593*.

A Appendix

A.1 Dataset Annotation Guidelines and Process

Before annotating the dataset at full-scale, we conducted a pilot study with the annotators on a small evaluation set that contains 10 document and summary pairs. We then discussed with the annotators and had them explain the labels they were given to ensure they fully understood the task and followed the guidelines. The guidelines can be summarized as follows:

(1) Read the source documentation and generated abstract. If the article is incomprehensible (e.g. too short or in a language other than English), mark it as corrupted.

(2) For each entity in the summary (identified by NER tool), check whether the entity can be directly entailed in the summary context using only the information within the source document. If the answer if yes, label the entity as non-hallucinated. If the entity has multiple properties, annotate each property separately.

(3) If the source does not contain sufficient information to entail the entity, use Wikipedia or Google Search to determine the factuality of the entity. If no information can be found to prove or disprove the factuality of the entity. Label it as non-factual hallucination.

(4) If the entity is mentioned in the source document, but it is used in the wrong context and misrepresents information from the document. Label the entity as intrinsic hallucination.

We also ask the annotators to mark and annotate entities missed by automatic NER tools. We will then update the identified entities to ensure that the samples are consistent for all annotators. Annotators are paid 20$ an hour for their work, which is above the minimum wage in their country of residence.

A.2 Patterns of Annotated Entities

Table 9 shows the patterns of hallucinated entities. For factual hallucinations, Person, GPE, and ORG are the three most common types. Among non-factual hallucinations, Date is the most common type (31.65%). Cardinal numbers are also easily hallucinated by summarization model. Note that the proportion of Date and GPE type of entities in non-factual hallucinations is much higher than their proportion in all entities.

	All	Factual hal.	Non-factual hal.
Person	30.16%	33.23%	20.25%
GPE	21.84%	21.75%	8.54%
ORG	15.03%	18.43%	7.91%
Date	11.32%	9.06%	31.65%
Cardinal	6.34%	3.63%	12.97%
Other	15.31%	13.90%	18.68%

Table 9: Percentage of each type of entity in the XENT dataset. GPE stands for geopolitical entity, i.e. countries, cities, states. ORG includes companies, agencies, institutions.

A.3 Experimental Setup

Dataset We use both XSUM, Narayan et al. (2018b)) and the CNN/Dailymail dataset (Hermann et al., 2015) in this work. CNN/DailyMail is a widely used summarization benchmark with 287,227 training samples, 13,368 validation samples, and 11,490 test samples. XSUM dataset contains 226,711 British Broadcasting Corporation (BBC) articles. Each article is paired with a single sentence summary written by the BBC journalists. The dataset is split into three subsets: training (204,045, 90%), validation (11,332, 5%), and test (11,334, 5%) sets.

Language Model Hyperparameters All language models used in this paper are based on the Transformer encoder-decoder architecture from the Fairseq library (Ott et al., 2019) that is written in PyTorch (Paszke et al., 2017). For the CMLM training, we initialize the model with the checkpoint of the large BART model. The max sequence length is set to 1024 for both the encoder and decoder modules. We fine-tuned the model for 15,000 steps with the warm-up steps set to 500. We use the standard cross-entropy loss as our objective function with 0.1 label-smoothing (Szegedy et al., 2016). The Adam optimizer (Kingma and Ba, 2015) with $\epsilon =$ 1e-8 and an initial learning rate 3e-5 are used for training. The dropout rate in each layer is set to 0.1. These hyperparameter values are based on the recommended values from the fairseq (Ott et al., 2019) library All experiments are conducted on 4 Tesla V100 GPUs with 32GB of memory.

RL Training In the off-line RL experiment, we initialize the model using the BART large model finetuned on XSUM dataset[4]. The discount factor γ is set to 1 and the learning rate r is set to $1e-5$.

[4] https://github.com/pytorch/fairseq/tree/master/examples/bart

We update the model for 30,000 steps in total with 1000 warm-up steps. We use polynomial decay to update the learning rate after each training step. No reward-shaping is used.

To make the training more stable, we use another policy network $\tilde{\pi}_\theta$ to compute the importance weight w. $\tilde{\pi}_\theta$ is kept as a slow copy of π_θ with the same model architecture. We use *Polyak updates* to slowly update the weight of $\tilde{\pi}_\theta$ in the direction to match π_θ every step. The update rate of $\tilde{\pi}_\theta$ is set to 0.01.

A.4 Classification Results on XENT Dataset

	Prec.	Recall	F1
Non-hallucinated	97.88	92.38	95.05
Factual hal.	60.84	84.87	70.88
Non-factual hal.	71.43	56.18	62.89

Table 10: Evaluation results on XENT. We report the leave-one-out error of our ENTFA model with prior, posterior probability and word overlap as features.

Table 10 shows the three-class classification results of our model on XENT dataset. Since we are the first work (to the best of our knowledge) that distinguishes between factual and non-factual hallucinations, we did not have a baseline model to compare with right now. We compare with other models separately in terms of factuality and hallucination classification in Section 6.1.

A.5 Prior/Posterior Probabilities

Figure 2 plots entities in the XENT dataset according to their prior and posterior probabilities and shows the KNN classification boundaries of ENTFA w/o overlap. In Figure 2a, we find that the non-factual hallucinated entities are clustered around the origin. This is in line with our expectations since non-factual hallucinations have lower prior and posterior probabilities. Both factual hallucinated and non-hallucinated entities are gathered in the top area with high posterior probabilities.

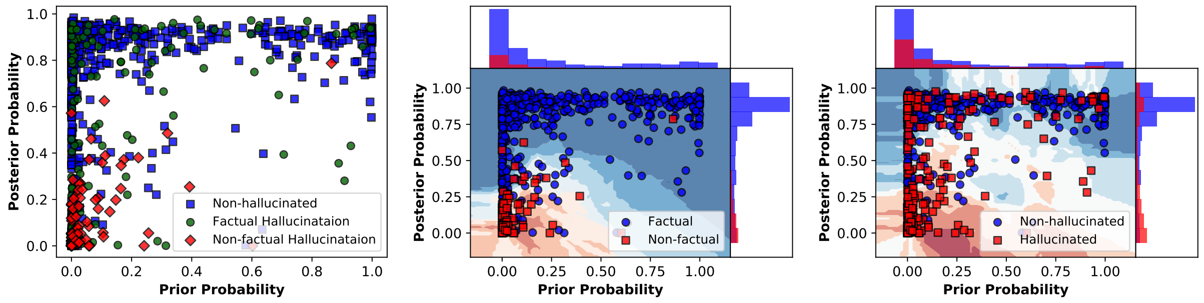
In Figure 2b, the KNN classifier separates the factual and non-factual entities with clear boundaries. A large part of the factual hallucinated entities are correctly identified by CMLM$_{\text{XSUM}}$ with relatively high posterior probabilities. This explains our model's superior performance on factuality checking. The top and right histograms in Figure 2b show the entity distribution over prior and posterior probability value respectively. As shown in 2b's histogram, factual entities have significantly higher posterior probability than that of non-factual entities on average.

Figure 3 shows histograms of the prior and posterior probabilities of entities from MLM and CMLM$_{\text{XSUM}}$, separated by their class (i.e., whether they are hallucinated and/or factual). Non-hallucinated entities have higher posterior probability than factual and non-factual hallucinations on average. The average posterior probability for non-hallucination, factual hallucinations, and non-factual hallucinations are 0.763, 0.599, and 0.133 respectively.

A.6 Evaluating Entity Factuality on Noisy Training Data

Recent work (Narayan et al., 2021; Nan et al., 2021) has shown that filtering out noisy training samples in the XSUM dataset can mitigate the hallucination issue. Therefore, we divide the XSum training set into clean samples and potentially noisy samples. Potentially noisy samples are samples where the reference summary contains entities that does not appear in the source. This gives us around 150k potentially noisy training samples and 50k clean training samples. Then, we mix the clean samples with noisy samples at different proportions to create training sets with different levels of noise. Figure 4 shows the evaluation results of summarization models trained on these datasets. We can see that the model generates fewer factual entities as the training set gets noisier. Also, it shows that ROUGE score is not a favorable metric in terms of factuality evaluation. Since with the training set size fixed, the model seems to achieve higher ROUGE score at the expense of entity factuality. In addition, this indicates that if the system is optimized only for ROUGE, they may inadvertently harm factual consistency.

We also observe that the word overlap method predicts much lower entity factuality rate than ENTFA. This is due to the fact that the word overlap method cannot identify factual hallucinations and introduce many false-negative samples. To verify this, we extracted all entities from summaries generated by the model trained on 50k noisy samples (x-axis = 1.0). Among these entities, there are 7,358 entities that do not appear in the source but are predicted as factual by our model. We find that 50.5% of these entities can be found in the ref-



(a)

(b)

(c)

Figure 2: The distribution of entities over prior/posterior probability. Each point in the figure represents an entity $(p_{\text{prior}}(e_k), p_{\text{pos}}(e_k))$ and shading indicates the confidence of the classifier. (a) The distribution of entities; (b) The entity factuality classification results with KNN ($k = 20$) classifier. Both factual hallucinated and non-hallucinated entities are colored blue; (c) The KNN ($k = 20$) classification boundaries of hallucinated and non-hallucinated entities.

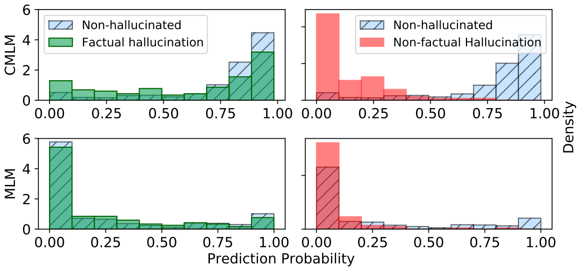


Figure 3: Normalized histogram of model prediction probability for three classes of entities. The first row shows the entities' posterior probability calculated using CMLM. The second row shows the prior probability from MLM.

erence summary. As a contrast, only 12.7% entities predicted as non-factual by our model can be found in the reference.

Figure 5 shows the evaluation result of PEGASUS model (Zhang et al., 2020) follows the evaluation set up in Section A.6. Both figures show a similar trend that the models get higher ROUGE score when trained on noisier dataset with the cost of generating more non-factual entities.

Compared with BART model, PEGASUS generates more hallucinated entities and has higher ROUGE score overall. For instance, when both trained on 50k clean data, PEGASUS has ROUGE-1 score 0.450 compared with BART's 0.406. The predicted factual entity rate for PEGASUS and BART is 84.79% and 91.81% respectively. This may be due to the fact that PEGASUS is pre-trained on a much larger corpus than BART. We leave the study of this phenomenon to future work.

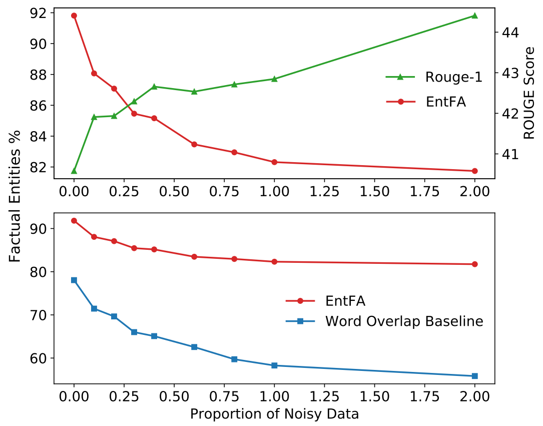


Figure 4: Evaluation of an abstractive summarization model (BART) trained on datasets with different levels of noise. The y-axis on the left represents the percentage of factual entities classified as factual by (ENTFA) or the word overlap baseline. The y-axis on the right indicates ROUGE-1 scores. X-axis = 0 and x-axis = 1.0 means that the model is trained on 50k clean samples and 50k noisy samples respectively; x-axis = 0.5 represents the model trained on a mix of 25k clean samples and 25k noisy samples. X-axis = 2.0 represents a model that is trained on 100k noisy samples. All models are tested on XSUM's official test set. We observe a similar trend with the PEGASUS model (Figure 5).

A.7 Where Does the Model Learn to Hallucinate?

Table 3 shows that 30% of the entities in the summaries generated by BART are hallucinated, including 15% factual hallucinated entities. To generate factual hallucinated entities, the summarization model needs to integrate background knowledge into the summary. One interesting problem is investigate where the model learns that knowledge. Since the BART is pre-trained on a large text corpus and fine-tuned on XSUM, the knowledge of

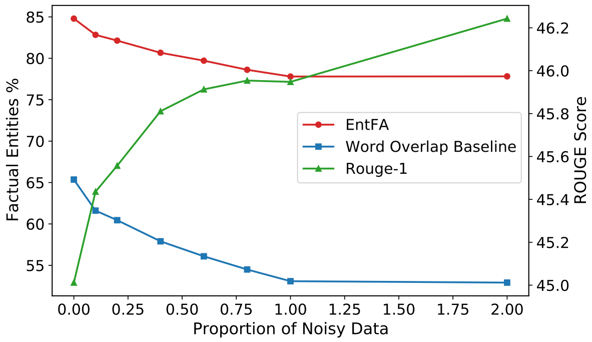


Figure 5: Evaluation of PEGASUS$_{\text{LARGE}}$ trained on datasets with different levels of noises.

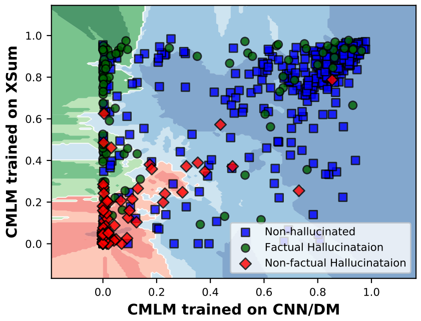


Figure 6: Entity distribution over posterior probabilities from CMLM$_{\text{XSUM}}$ and CMLM$_{\text{CNN/DM}}$. The shading shows the classification boundaries of the classifier.

hallucinated entities could come from either the pre-training corpus or the XSUM training set. To investigate this, we trained a separate CMLM on the CNN/DM dataset.

Figure 6 shows the entity distribution from the two CMLM models. For non-hallucinated entities, the distributions are similar; for factual hallucinations, we can find that a large portion of them has very low posterior probabilities under CMLM$_{\text{CNN/DM}}$, but high posterior under CMLM$_{\text{XSUM}}$. This pattern suggests that the knowledge of many factual hallucinations comes from the XSUM training set.

We define $\sigma(e_k) = \log \frac{P_{\text{CMLM}_{\text{XSUM}}}(e_k)}{P_{\text{CMLM}_{\text{CNN/DM}}}(e_k)}$. If $\sigma(e_k) \geq 0$, it suggests that CMLM$_{\text{XSUM}}$ is more confident that e_k is factual than CMLM$_{\text{CNN/DM}}$. For a factual hallucination e_k, we can infer that the knowledge of e_k is in XSUM if $\sigma(e_k)$ is large. To further verify this, we retrieve the 10 most similar documents from XSUM and CNN/DM for each factual hallucinated entity using TF-IDF. Then, we count the number of times each entity appears in those similar training samples. For entities with $\sigma(e_k) \geq 5$, the average number of appearances is 2.19 on XSUM and 0.77 on CNN/DM. For entities with $\sigma(e_k) \leq 0$, the average number of appearances becomes 2.85 and 2.46 on XSUM and CNN/DM respectively. This further confirms that the knowledge of factual hallucinations with large $\sigma(e_k)$ comes from XSUM.

A.8 Compare with Filippova (2020)'s Work

Filippova (2020)'s work on data-to-text generation shows that low posterior probability from a CLM during decoding indicates hallucination. Take the summarization model as an example, if an entity is generated with very low posterior probability, it is likely that the generated entity is hallucinated and non-factual. However, compared with CMLM, CLM has more uncertainty during decoding since the right context of the entity is not determined. The uncertainty of the CLM comes from both content selection (text content and structure) and lexical choice (Xu et al., 2020). For CMLM though, the uncertainty is mostly reduced to the latter.

Figure 7 show the entity posterior probabilities from CLM and CMLM model. As shown in the figure, we can find that most factual entities (blue points) are above the $x = y$ line. This means CMLM gives more certainty to the same factual entity than CLM. The ROC curve in Figure 8 further shows this. As the lines get closer to the origin, the threshold becomes larger, and CMLM has a higher TPR than CLM. This means CMLM will classify more entities as factual. The higher AUC value of CMLM further demonstrates that CMLM is a better choice for factuality checking than CLM.

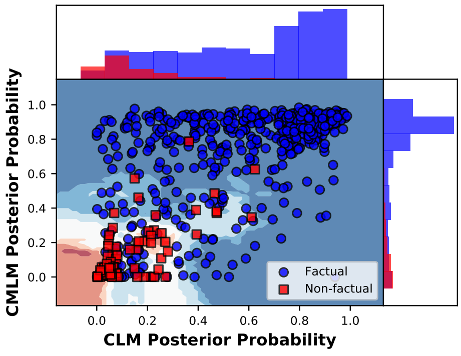


Figure 7: Posterior probabilities calculated from CLM and CMLM. Both models are trained on XSUM dataset.

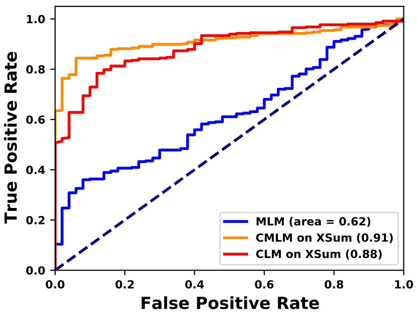


Figure 8: ROC curve of entity's posterior probability and factuality.