DETECTING HALLUCINATED CONTENT IN CONDITIONAL NEURAL SEQUENCE GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural sequence models can generate highly fluent sentences but recent studies have also shown that they are also prone to hallucinate additional content not supported by the input, which can cause a lack of trust in the model. To better assess the faithfulness of the machine outputs, we propose a new task to predict whether each token in the output sequence is hallucinated conditioned on the source input, and collect new manually annotated evaluation sets for this task. We also introduce a novel method for learning to model hallucination detection, based on pretrained language models fine tuned on synthetic data that includes automatically inserted hallucinations. Experiments on machine translation and abstract text summarization demonstrate the effectiveness of our proposed approach – we obtain an average F1 of around 0.6 across all the benchmark datasets and achieve significant improvements in sentence-level hallucination scoring compared to baseline methods. We will also release our annotated data and code for future research.

1 INTRODUCTION

Neural sequence models have achieved impressive breakthroughs in a wide range of applications, including free-form question answering (Fan et al., 2019), machine translation (Vaswani et al., 2017; Wu et al., 2016) and text summarization (Rothe et al., 2020). Although these models can generate fluent sentences that are even sometimes *preferred* to human-written content (Läubli et al., 2018; Brown et al., 2020), recent work has also shown that they lack global logical consistency (Marcus & Davis, 2020), sometimes degenerate to dull and repetitive outputs (Welleck et al., 2019) and can often hallucinate content that is not entailed by the input (Maynez et al., 2020). In this paper, we focus on the faithfulness of machine outputs in conditional sequence generation tasks, aiming to automatically identify and quantify content in the output that is not faithful to the input text.

This risk of generating unfaithful content impedes the safe deployment of neural sequence generation models. The first step to building models that do not suffer from these failures is the assessment and identification of such hallucinated outputs. Prior work has shown that standard metrics used for sequence evaluation (Maynez et al., 2020; Wang & Sennrich, 2020) do not correlate well with the faithfulness of model outputs, and very recent efforts (Maynez et al., 2020; Durmus et al., 2020; Wang et al., 2020a) have started to develop automatic metrics to measure the faithfulness of output sequences. These methods use an external semantic model, e.g. for question-answering or textual entailment, to score faithfulness. However, these scores do not directly measure the number of hallucinated tokens and only correlate weakly with human judgements due to compounded errors.

We propose a new task for faithfulness assessment - hallucination detection at the token level, which aims to predict if each token in the machine output is a hallucinated or faithful to the source input. This task does not use the reference output to assess faithfulness, which offers us the ability to apply it in the online generation scenario where references are not available. We measure hallucination for two conditional sequence generation tasks – abstractive summarization and machine translation (MT). For the former, we produce a benchmark dataset from the recent released annotations in (Maynez et al., 2020). For MT, we carefully design the human assessment guideline and create high-quality annotations. We will also release our human annotated data for future research.

To learn token-level hallucination prediction for general conditional sequence generations tasks, we propose a novel method that creates synthetic "hallucinated" dataset with pseudo labels and finetunes

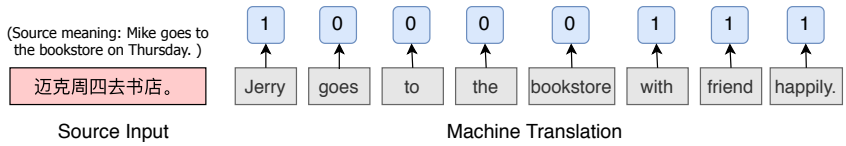

Figure 1: A toy example of token-level hallucination detection from machine translation. The words in grey blocks is an example of machine translation output and the labels above them indicate if each word is faithful (0) to the source input or a hallucinated one (1).

a pretrained language model (Liu et al., 2019; Conneau et al., 2020) on it. Without any human annotated supervised training data, we achieve an average F1 of around 0.6 across all the benchmark datasets, setting initial performance levels for this new task. We have also computed sentence-level aggregated predictions and achieve significantly higher correlations with human scores than previous methods. Finally, we use our new data to study the effect of pretraining on MT hallucination and show it can actually produce more faithful translations, confirming recent findings in abstractive summarization (Maynez et al., 2020).

2 Related Work

Commonly adopted metrics for sequence generations evaluations such as BLEU scores (Papineni et al., 2002; Post, 2018), ROUGE (Lin & Hovy, 2004) and recently proposed quality estimation metrics including BLEURT (Sellam et al., 2020) and BERTScores (Zhang et al., 2019) have been shown that they correlate poorly with faithfulness (Maynez et al., 2020; Wang & Sennrich, 2020). They also require reference text, limiting their applicability to detecting halluciations in a deployed system at run-time.

Recently, several efforts have been made on automatic evaluation of faithfulness and truthfulness in abstract text summarization (Wang et al., 2020a; Maynez et al., 2020; Durmus et al., 2020; Falke et al., 2019) by leveraging existing semantic inference systems: (1) **question answering (QA) system**-based methods (Wang et al., 2020a; Durmus et al., 2020) first employ a question generation system to generate natural questions based on the machine generated summary, then questions are answered with a QA system using the source article and the summary respectively. The final faithfulness score of the summary is given by comparing the corresponding answers. (2) For the **textual entailment**-based methods, a natural language inference (NLI) system is trained to predict whether the source article entails the summary or not. Then the entailment scores are used as the proxy to assess the faithfulness of the summary. Nonetheless, these methods tailored for abstract text summarization are shown to have a weak correlation with human evaluation of faithfulness.

Similar to the spirit of our proposed task, word-level quality estimation (Fonseca et al., 2019) in the machine translation community predicts if tokens are correctly translated based on human post-editing. However, they do not distinguish errors in terms of fluency and adequacy (Specia et al., 2011). In contrast to estimating the amount of human post-editing work required to fix errors, we specifically identify hallucinations corresponding to only hallucination (not fluency) errors.

3 Task: Hallucination Prediction at Token-level

Our token-level hallucination detection task builds on the definition of hallucination introduced by Maynez et al. (2020). For a source sequence S and its model generation G from a neural conditional generation model, we define any span $w_i, \cdots, w_{i+j} (j >= 0)$ in G as being hallucinated if it cannot be entailed by the source input S. More specifically, we consider two types of hallucination:

Content Insertion: a span $w_i, \cdots, w_{i+j}$ in G consists of additional content that is not supported by S, i.e. its paraphrase or other equivalent form cannot be inferred from S. In Fig. 1, the word *"happily"* in the machine translation belongs to this case. This is also referred as "extrinsic hallucinations" in Maynez et al. (2020).

Incorrect Substitution: a span of word(s) is misrepresented information based on the S. In Fig. 1, *"Jerry"* in the machine translation is a hallucinated word and should be replaced by *"Mike"*. This

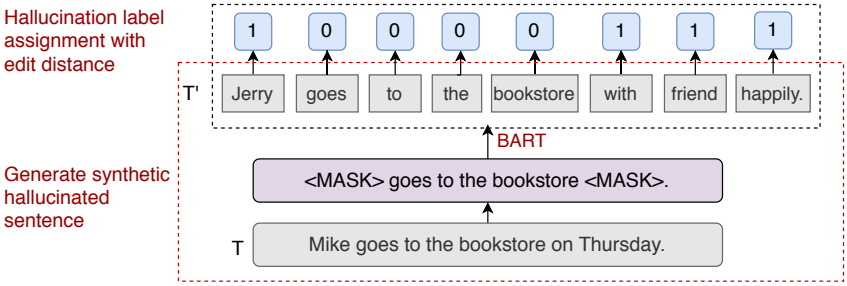

Figure 2: Generation of synthetic hallucinated data set with hallucination labels. The bottom block generates a hallucinated version of T by feeding the noised sentence to the pretrained encoder-decoder model BART. The top block assigns hallucination labels to each token by computing the edit-distance between T' and T. Labels of 1 refer to hallucinated words and vice versa.

type of hallucination is similar to the concept of "intrinsic hallucination" in Maynez et al. (2020). Note that there are cases where certain words (e.g. "This is not a great book." becomes "This is a great book.") are dropped in G and hence the meaning of S is changed, and we consider any spans in G that misrepresent S as hallucinated contents (e.g. the entire sentence of "This is a great book.").

We aim to identify all the span(s) satisfying the above conditions in the machine generation G.[1] Such fine-grained predictions of hallucination can not only provide the faithfulness level of the entire sentence by aggregating token-level predictions but also offer benefits for more general use cases by labeling specific hallucinated tokens (e.g. interpretability of generations).

4 Learning Token-level Hallucination Detection

We propose a general-purpose method for token-level hallucination detection for conditional sequence generation tasks. Given the source input S, we first formulate the task of token-level hallucination detection as a sequence labeling problem where a binary label is predicted at each position G_t of the machine generation G. One straightforward way of learning such task is to train a model with supervised data in the form of $((S, G), L_G)$ where L_G are the labels at every positions of G that indicate if each word is a hallucinated one or not. However, because such labeled training data is not readily available, we also propose a approach to automatically create synthetic training data.

4.1 Synthetic Data Creation

We use bi-text from the training data to create synthetic examples by automatically inserting new, hallucinated tokens. More speficially, we take target sequence T and create a hallucinated version of it denoted T' with associated hallucination labels for each token in T'. Then we can train a supervised model on this synthetic labeled data set of $((S, T'), L_{T'})$. The key challenge is that T' should be a fluent sentence that does not differ too much from T.

Generation of hallucinated sentences To control this synthetic hallucination process, we build on recent work on large scale denoising autoencoders. These models learn to map a corrupted sentence back to the original text it was derived from, including for example learning to reconstruct missing words that have been arbitrarily masked out. We use BART (Lewis et al., 2020) by first inserting noises and then reconstructing parts of the target sentence, without providing it any access to the source sentence, thereby encouraging it to insert new content as needed to ensure fluency. As shown in Fig. 2, we first apply noising functions to the target sentence T in the bi-text and then use a pretrained BART to generate T' conditioned on the noised T with beam search.

Label assignments After obtaining the hallucinated sentence T' with BART, we need to assign appropriate labels to each token in T' to mark which words are hallucinated. We compute the edit distance between T' and T, and back-trace the deletion and substitution operations with dynamic programming. All the positions in T' involving these two operations are labeled as hallucinations and everything else is considered faithful to T.

[1] We do not consider the under-generations e.g. the source input is only partially translated or summarized.

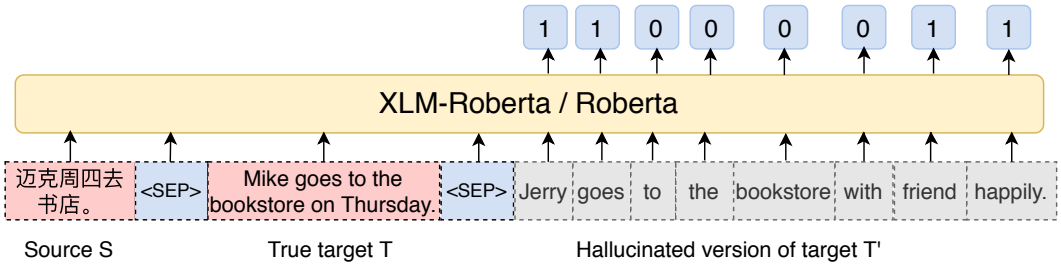

Figure 4: Finetuning XLM-Roberta (for cross-lingual generation task, e.g. machine translation) or Roberta (for monolingual generation task, e.g. text summarization) on the synthetic training data.

Fig. 3 shows an example of label assignment with edit distance, where words in red are replaced and the word in blue is deleted to convert T' to T. Assigning labels with edit-distance can not always guarantee the correct labels, however, we find that this simple approach provides sufficiently high quality training data for effective hallucination detection in practice.

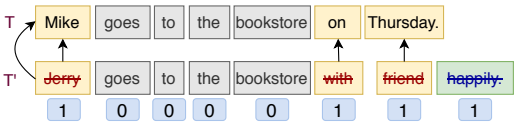

Figure 3: An example of label assignment.

4.2 Finetuning Pretrained LM on Synthetic Data

Hallucination prediction loss We follow the common practice in natural language understanding (NLU) tasks and finetune a pretrained language model (LM) on our synthetic data. We finetune a cross-lingual LM (Conneau et al., 2020) for machine translation and a monolingual model (Liu et al., 2019) for summarization. In both cases, we concatenate the input source, true target and hallucinated target denoted (S, T, T') as a single input sequence to the pretrained LM. Then we minimize the standard classification loss $\mathcal{L}_{pred}$ over the pseudo hallucination labels $L_{T'}$ on top of the final hidden vectors of each token in T' as shown in Fig. 4.

Although using only the source text and hallucinated target (S, T') as the input should be sufficient to learn the hallucination prediction task, we can also easily measure the extent to which including the true target T in the input could help the model. At test time, when evaluating the faithfulness of the machine outputs G, we do not use the true target T and perhaps surprisingly find our model can generalize well without references, even when it was present during training.

To prevent the model from overly relying on the true target T and learning spurious correlations (e.g. the edit distance), we explored two techniques: (1) *dropout* – randomly drop out tokens in T to force the dependence on the source input; (2) *paraphrase* – recall that at synthetic data generation time, we generate T' from BART conditioned on the noised T. Instead, we can apply noise functions to the paraphrased sentence of T. Let D denote the paraphrased sentence of T and D' denote the generation from BART conditioned on the noised D. Then we create pseudo labels of D' denoted $L_{D'}$ by computing the edit-distance between the D' and D and use $((S, T, D'), L_{D'})$ as the training data for finetuning. Since the pseudo labels are created based on D, it can prevent the model from learning the edit-distance between T and D' easily. Ablation studies will be provided in § 6.4.

Masked LM loss We also add the masked language model loss (MLM) $\mathcal{L}_{mlm}$ following (Devlin et al., 2019). To learn this loss, we create a different batch from the above by concatenating only the source S and target T as the input, since the hallucinated target T' could provide erroneous information for predicting masked words in T. We find that such multi-task learning objective helps learn better representations of the inputs and further improves performance on predicting hallucination labels. The final loss is $\mathcal{L} = \mathcal{L}_{pred} + \alpha \cdot \mathcal{L}_{mlm}$ where α is a hyperparameter.

5 Evaluation Tasks and Data

We evaluate hallucination in conditional neural sequence models on abstractive text summarization and machine translation (MT) tasks, using the models and datasets described below.

5.1 ABSTRACTIVE TEXT SUMMARIZATION

Maynez et al. (2020) studied the hallucination problems in extreme summarization on the XSUM dataset which comprises 226,711 British Broadcasting Corporation (BBC) articles paired with their single-sentence summaries. They randomly sampled 500 articles from the test set of XSUM and evaluated summaries from four abstractive summarization systems: (1) **PtGen** (See et al., 2017), an RNN-based attentional sequence-to-sequence (Seq2Seq) model. (2) **TConvS2S** (Narayan et al., 2018) is a topic-aware convolutional (Gehring et al., 2017) Seq2Seq model. (3) **TranS2S** and (4) **BERTS2S** are standard Transformer (Vaswani et al., 2017) encoder-decoder models. The weights in TranS2S are randomly initialized while the encoder and decoder of BERTS2S are initialized with the pretrained BERT-base (Devlin et al., 2019) checkpoint. Maynez et al. (2020) asked the human annotators to label the spans in the machine generated summaries if they were unfaithful to the article (for more details see Maynez et al. (2020)). We post-processed their human annotations by majority voting and create our test datasets for each of the abstractive summarization systems.

5.2 MACHINE TRANSLATION

Previous work (Wang & Sennrich, 2020; Müller et al., 2019; Koehn & Knowles, 2017) has shown that translation models are particularly prone to hallucination when tested out of domain. We similarly focus on this evaluation regime and additionally consider the low resource case where a modest amount of out of domain data is available at training time.

Data We use a multi-domain Chinese-English (Zh-En) translation dataset (Wang et al., 2020b) which consists of four balanced domains: *law, news, patent* and *subtitles*. We create a new training data $\mathcal{D}_{train}$ with *law* (1.46M sentences), *news* (1.54M), *subtitles* (1.77M) train data and randomly sample 0.03% parallel sentences from the *patent* (870) training data. We train two NMT models (described below) on this dataset and test on 150 examples sampled from patent test data. In addition, we also evaluate the trained models on the COVID-19 domain. We sample 100 examples from the recently released translation benchmark dataset (Anastasopoulos et al., 2020). Together with the 150 examples from patent test data, we create our benchmark dataset with 250 examples in total and we denote this dataset $\mathcal{D}_{eval}$. We ask human annotators to evaluate the level of hallucinations on this dataset $\mathcal{D}_{eval}$.

Models	**Fleiss' Kappa**	
	Token	**Sent**
MT		
TranS2S	0.58	0.72
MBART	0.54	0.62
XSum		
PtGen	0.81	-
TConvS2S	0.83	-
TranS2S	0.79	-
BERTS2S	0.79	-

Table 1: Fleiss's Kappa scores (↑): agreements on **token**-level hallucination labels or sentence-level (**sent**) ratings among different annotators. The token-level agreements for XSUM are computed on the released annotations by Maynez et al. (2020).

Models Our evaluation data is generated from two reference models on which we will measure hallucination (see Appendix for more details): (1) **TranS2S** (Vaswani et al., 2017) is the standard Transformer Seq2Seq model with 6 encoder layers and 6 decoder layers. (2) **MBART** (Liu et al., 2020) is a Seq2Seq denoising auto-encoder pretrained on large-scale monolingual corpora in many languages. We finetune a MBART checkpoint on $\mathcal{D}_{train}$ which is a standard Transformer architecture with 12 layers of encoder and 12 layers of decoder pretrained on a multilingual corpora of 100 languages.

5.3 HUMAN ASSESSMENT OF HALLUCINATIONS

In the human evaluation, three bilingual annotators were presented the Chinese source sentence, the English reference sentence and the machine translation. We shuffled the orders of sentences so that annotators did not know which translation model was used (TranS2S or MBART).

We conducted the pilot study and practice sessions with annotators before annotating the blind test set $\mathcal{D}_{eval}$. Annotators were asked to only focus on the hallucinations during assessment and label each sentence with one of the three types of labels: incomprehensible, faithful, and hallucinated. If the translation is a hallucinated one, we asked the annotators to tag all the tokens that were not faithful to the source. We dropped all the translations that were labeled as incomprehensible. The final benchmark datasets were created by taking majority labels among three annotators. (See Appendix A for more annotation details.)

Tasks	Models	test w/o reference			test w/ reference		
		Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**
MT	TranS2S	64.27	67.30	65.75	59.52	74.27	66.08
	MBART	49.56	36.32	41.92	43.13	53.63	46.81
Summarization	PtGen	57.83	69.92	63.30	56.51	73.48	63.89
	TConvS2S	63.56	67.75	65.59	61.68	71.63	66.28
	TranS2S	57.02	67.23	61.70	55.85	69.65	61.99
	BERTS2S	49.83	62.50	55.45	48.39	66.11	55.88

Table 2: Precision, recall and F1 (x100) of hallucination labels on the outputs of different systems from machine translation on a Zh-En multi-domain dataset (see§5.2) and abstract summarization on the XSUM dataset.

In Tab. 1, we show the Fleiss's Kappa scores of our annotations for MT and the processed annotations from (Maynez et al., 2020) on abstractive summarization. A higher Fleiss's Kappa score indicates higher agreements among annotators.[2]. We achieved moderate agreement on the token-level hallucination annotations and substantial agreement on the sentence-level annotations, while Maynez et al. (2020) achieved substantial or almost perfect agreement on the XSUM dataset. For MT, it's relatively harder to achieve consistent agreements among annotators for the following reasons: first, although we have made strict annotation guidelines following the definition of hallucination in Section 3, it could still be difficult for annotators to distinguish between bad translations and hallucination; second, it was sometimes difficult for annotations to understand the specialized text in the patent domain, which can contain complex scientific descriptions.

6 EXPERIMENTS

6.1 EXPERIMENTAL SETUP

Synthetic Data Generation We use a pretrained BART (Lewis et al., 2020) model in the fairseq toolkit (Ott et al., 2019) with 12 layers of encoder and 12 layers of decoder for synthetic labeled data generation. We uniformly sample the percentage of tokens p_m to mask from $[0, h_m]$ for each sentence. We also uniformly sample the probability of replacing a token with a random token from $[0, h_r]$ denoted p_r. p_m and p_r are two important factors that affect the noise level when generating the synthetic data. For MT, we set h_m and h_r to 0.6 and 0.3 respectively. For abstractive summarization, we use 0.4 and 0.2. We use beam search for decoding from BART with beam size of 4 and length penalty of 3. For MT, we first create paraphrased target sentences D' through knowledge distillation (Kim & Rush, 2016) by using the outputs from the same trained TranS2S model on the source inputs.

Finetuning Pretrained LM For MT, we finetune XLM-R (Conneau et al., 2020) on the synthetic dataset with batch size of 128, and we annotated 50 examples (different from those in $\mathcal{D}_{eval}$) from the patent test data as the validation dataset. For summarization, we finetune RoBERTa (Liu et al., 2019) with batch size of 96 and early stop training with 10K update steps. In addition, we dropout tokens from the reference T in the input with a rate of 0.5 and 0.3 respectively for summarization and MT to learn $\mathcal{L}_{pred}$. We set α to be 0.6 for MT and 0.5 for summarization based on the scales of $\mathcal{L}_{pred}$ and $\mathcal{L}_{mlm}$. For both tasks, we set the mask probability used for $\mathcal{L}_{mlm}$ to be 0.3, and the initial learning rate to be $2e-5$ with polynomial decay. We describe other hyperparameters, including training of MT models, in the Appendix B.

6.2 RESULTS

Evaluation on token-level hallucination prediction In Tab. 2, we present the precision, recall and F1 of token-level hallucination labels across six benchmark datasets for MT and astractive summarization. We achieve descent performance on this task, however it is still far from being solved. Token-level hallucination prediction requires deep semantic understanding of the inputs and is worthy of study in the future. We can see that even though our model learns hallucination prediction with reference T during training (Sec. 4.2), by applying token dropout to T, our model generalizes

[2]https://en.wikipedia.org/wiki/Fleiss%27_kappa

Models	Entail	Align	Ours (P)	Ours (R)
MT				
TranS2S	-0.32	0.23	0.78	**0.79**
MBART	-0.19	0.05	0.36	**0.48**
Summarization				
PtGen	0.25	0.32	**0.42**	0.37
TConvS2S	0.26	0.31	**0.42**	0.39
TranS2S	0.13	0.19	**0.35**	0.33
BERTS2S	0.23	0.27	**0.36**	0.36

Table 3: Spearman's correlation coefficients of the sentence level hallucination scores computed by different systems with the human annotations (percentage of hallucinated tokens in a sentence). **Entail** represents "textual entailment" and **Align** represents "alignment model between model generation and the source" (see 6.2).

Input to $\mathcal{N}(\cdot)$	Precision	Recall	F1
MT			
raw	58.35	70.12	63.70
TranS2S distill	64.27	67.30	65.75
Summarization			
raw	57.02	67.23	61.70
Extractive distill	54.10	36.45	43.55
Abstractive distill	57.33	28.59	38.16

Table 4: Performance on the TranS2S benchmark from MT and summarization by using different data as the input to the noised function $\mathcal{N}(\cdot)$. "raw" refers to the original targets in the training data.

well without feeding the reference at test time. In contrast, we include the reference at test time and observe that the model can achieve a significantly higher recall but worse precision. Although the proposed method performs relatively worse on the benchmark data of MBART, it performs surprisingly well on other data sets given that we did not use any real annotations at training time.

Comparisons on sentence-level hallucination scores We aggregate the token-level predictions to score the level of hallucination of each sentence and compute the Spearman's correlation coefficients between our scores and the human annotations of percentage of hallucinated tokens in a sentence. In Tab. 3, we present two scores from our model: (1) Probability-based score (**P**) computes the score by averaging the halluciation probabilities across all the tokes in G. (2) Ratio-based score (**R**) first predicts the hard labels by taking $\arg\max$ at each position and then use the percentage of hallucinated labels as the score.

We compare with two baseline metrics that do not require reference as well: (1) Textual entailment prediction model (**Entail**) has been proposed to evaluate the faithfulness of abstractive summaries (Maynez et al., 2020; Kryściński et al., 2019; Falke et al., 2019). We adapt it to MT by finetuning XLM-Roberta on the machine translated Zh-En Multi-NLI dataset (Conneau et al., 2018). For summarization, we use the finetuned Roberta-large (Ott et al., 2019) on the Multi-NLI dataset (Williams et al., 2018). We calculate the entailment probability P_e between the source input and the machine output G and use $1 - P_e$ to score its hallucination level (which is better than use of contradiction probability in our experiments). (2) We propose to use a word alignment model (**Align**) for hallucination assessment. We employ SimAlign (Sabet et al., 2020), an unsupervised word aligner, that extracts word alignments from similarity matrices induced from pretrained word embeddings. SimAlign is essentially used for crosslingual tasks, and we adapt it to summarization by using embeddings from the pretrained BERT-large (Devlin et al., 2019). We compute the percentage of tokens in the machine output G that get aligned aligned to the source input to measure the faithfulness of G.

Overall, in Tab. 3 we see that: (1) For MT, we achieve very significant improvements over the baseline metrics. And both entailment and alignment scores correlate very poorly with the hallucination assessments. (2) For abstractive summarization, our metrics also significantly outperform baseline metrics. The alignment scores stand out as a better hallucination assessment compared to the commonly adopted entailment scores. We also note that for MT our ratio-based scores correlate better with human judgements while for abstractive summarization the probability-based scores have a better correlation. This shows that our model can produce more confident predictions for MT.

6.3 Analysis on Pretrained Models for Conditional Sequence Generation

Recent work (Maynez et al., 2020) has shown that pretrained models are better at generating faithful summaries as evaluated by humans. We confirmed this trend in machine translation. In Tab. 5, we present the detokenized BLEU scores, the percentage of hallucinated tokens in the evaluation

Models	BLEU	True hal. tokens (%)	Pred hal. tokens (%)
TranS2S	8.25	18.12	18.56
MBART	11.10	7.99	5.86

Table 5: Comparisons of TranS2S and MBART. **True hal. tokens** and **Pred hal. tokens** are the ground-truth and our model predicted percentage of hallucinated tokens in the benchmark test set.

set from human annotations and our model predictions. We can see that translations from MBART contain much less hallucinated contents than that from TranS2S. Moreover, our predictions match the human assessment very well at the corpus level.

6.4 Ablation Studies

Effects of including reference at training time Recall that we concatenate the source, reference and machine generation together as the input when learning hallucination predictions (Sec. 4.2). In Fig.5, we vary the dropout rate of tokens in the reference at training time and evaluate the models on the outputs from the TranS2S model for both tasks, where dropout rate of 1.0 indicates that we do not include the reference at all. **First**, different dropout rates do not signficinatly affect performance for MT, this is likely because we use the paraphrased target when creating the synthetic data instead of the reference sentences. Thus, the "hallucinated" sentences D' from BART do not resemble the reference T as closely as T', and the model will not learn spurious correlations between the T and D'. **Second**, for summarization we see that applying word dropout is crucial since we have used the reference more directly for generating synthetic data. On the other hand, if reference is removed at learning time (dropout = 1.0), the resulted model performs poorly, which shows that including reference at training time also has positive effects.

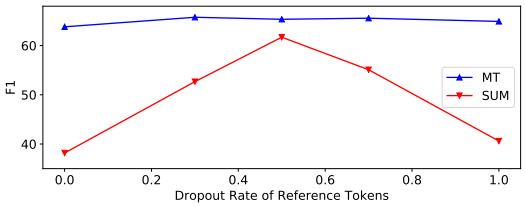

Figure 5: Performance on the TranS2S benchmark from MT and summarization by varying the dropout rate of tokens in the reference at learning hallucination predictions.

Effects of paraphrased data We investigate the effects of using paraphrased data in Tab. 4, where we apply the noise functions to different forms of targets when generating synthetic data. For MT, we create paraphrased targets via knowledge distillation (Kim & Rush, 2016) where we use the output from TranS2S conditioned on the source sentence in the bi-text corpus as the paraphrased target. We can see that with distillation data for synthetic data generation, the model achieves better results compared to using the references. However, note that we need to choose a proper word dropout rate when using the reference-based synthetic data as discussed above.

For abstractive summarization, we create paraphrased data out of an abstractive and an extractive summarization systems respectively. We finetune BART on the bi-text of XSUM and create distillation data from this finetuned abstractive model. For the extractive system, we use the recent proposed MatchSum (Zhong et al., 2020) as the distillation model. We see a significant drop in the performance for both of the variants. This likely due to the fact that: (1) it has been shown that abstractive summarization systems are prone to hallucinate contents themselves (Maynez et al., 2020), thus we are not able to create reliable pseudo labels based on the generated summaries, and (2) the extractive system generates summaries out of the input article which diverge from the actual abstractive summaries we evaluate on, and the model cannot generalize well under such data shift.

7 Conclusions

In this work, we proposed a new evaluation task for hallucination detection in conditional sequence generation and created human-annotated benchmark datasets. We also proposed a novel and general-purpose method to learn this task. In this future, we hope to create a large-scale pretrained evaluation model for any datasets or models to be evaluated. We also would like to extend our method to any data-to-text generation scenarios. We are also interested in investigating how to leverage our detection methods to mitigate hallucination problems in conditional sequence generation.

REFERENCES

Antonios Anastasopoulos, Alessandro Cattelan, Zi-Yi Dou, Marcello Federico, Christian Federman, Dmitriy Genzel, Francisco Guzmán, Junjie Hu, Macduff Hughes, Philipp Koehn, et al. Tico-19: the translation initiative for covid-19. *arXiv preprint arXiv:2007.01788*, 2020.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov. Xnli: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2018.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online, July 2020. URL https://www.aclweb.org/anthology/2020.acl-main.747.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*, 2019.

Esin Durmus, He He, and Mona Diab. 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*, Online, July 2020. URL https://www.aclweb.org/anthology/2020.acl-main.454.

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

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. Eli5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3558–3567, 2019.

Erick Fonseca, Lisa Yankovskaya, André F. T. Martins, Mark Fishel, and Christian Federmann. Findings of the WMT 2019 shared tasks on quality estimation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pp. 1–10, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-5401. URL https://www.aclweb.org/anthology/W19-5401.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1243–1252, 2017.

Yoon Kim and Alexander M Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1317–1327, 2016.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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

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

Samuel Läubli, Rico Sennrich, and Martin Volk. Has machine translation achieved human parity? a case for document-level evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 4791–4796, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1512. URL https://www.aclweb.org/anthology/D18-1512.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 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*, Online, July 2020. URL https://www.aclweb.org/anthology/2020.acl-main.703.

Chin-Yew Lin and Eduard Hovy. Rouge: A package for automatic evaluation of summaries. In Stan Szpakowicz Marie-Francine Moens (ed.), *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. Multilingual denoising pre-training for neural machine translation. 2020.

Ruixuan Luo, Jingjing Xu, Yi Zhang, Xuancheng Ren, and Xu Sun. Pkuseg: A toolkit for multi-domain chinese word segmentation. *CoRR*, abs/1906.11455, 2019. URL https://arxiv.org/abs/1906.11455.

Gary Marcus and Ernest Davis. Gpt-3, bloviator: Openai's language generator has no idea what it's talking about. *MIT Technology Review*, 2020.

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

Mathias Müller, Annette Rios, and Rico Sennrich. Domain robustness in neural machine translation. *arXiv preprint arXiv:1911.03109*, 2019.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 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, 2018.

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

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL 2002*, 2002.

Matt Post. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pp. 186–191, Belgium, Brussels, October 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W18-6319.

Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. Leveraging pre-trained checkpoints for sequence generation tasks. *Transactions of the Association for Computational Linguistics*, 8:264–280, 2020.

Masoud Jalili Sabet, Philipp Dufter, and Hinrich Schütze. Simalign: High quality word alignments without parallel training data using static and contextualized embeddings. *arXiv preprint arXiv:2004.08728*, 2020.

Abigail See, Peter J Liu, and Christopher D Manning. 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)*, pp. 1073–1083, 2017.

Thibault Sellam, Dipanjan Das, and Ankur P Parikh. Bleurt: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online, July 2020.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1162. URL https://www.aclweb.org/anthology/P16-1162.

Lucia Specia, Najeh Hajlaoui, Catalina Hallett, and Wilker Aziz. Predicting machine translation adequacy. In *Machine Translation Summit*, volume 13, pp. 19–23, 2011.

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

Alex Wang, Kyunghyun Cho, and Mike Lewis. Asking and answering questions to evaluate the factual consistency of summaries. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online, July 2020a. URL https://www.aclweb.org/anthology/2020.acl-main.450.

Chaojun Wang and Rico Sennrich. On exposure bias, hallucination and domain shift in neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online, 2020. URL https://www.aclweb.org/anthology/2020.acl-main.326.

Yong Wang, Longyue Wang, Shuming Shi, Victor OK Li, and Zhaopeng Tu. Go from the general to the particular: Multi-domain translation with domain transformation networks. In *Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI)*, New York, USA, February 2020b.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. Neural text generation with unlikelihood training. In *International Conference on Learning Representations*, 2019.

Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, New Orleans, Louisiana, June 2018. URL https://www.aclweb.org/anthology/N18-1101.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*, 2019.

Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. Extractive summarization as text matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online, July 2020.

A HUMAN EVALUATIONS

We asked three bilingual speakers to annotate the Chinese-to-English evaluation set $\mathcal{D}_{eval}$, which is composed of 150 sentences from the test set of Zh-En multi-domain dataset (Wang et al., 2020b), and 100 sentences from the COVID-19 translation benchmark dataset (Anastasopoulos et al., 2020) – TICO. TICO contains 6 finegrained domains including *Wikisource, Wikivoyage, Wikinews, CMU, PubMed* and *Wikipedia*. we randomly sample 25 examples from each of the four domains – Wikisource, Wikinews, CMU and PubMed, use these 100 samples for evaluation. We train standard base Transformer Seq2Seq and finetune the MBART (Liu et al., 2020) model on the training data from $\mathcal{D}_{train}$.

Annotation Guidelines We ask each annotator to read the tokens in the sentence carefully and check if they can be supported by the source sentence:

- If there are tokens (or the entire sentence) that cannot be supported by the source, label all the span(s) with color and mark the sentence as a hallucinated one;
- If the evaluator can not understand the entire translation at all, mark the sentence as incomprehensible;
- If all the tokens in the translation can be entailed from the source, mark the sentence as a faithful one.

Before the annotation work on the final blind test set, we first perform a pilot study on a different evaluation set and analyze the annotation results from each annotators. Then we conduct an education session to make sure that all the evaluators can fully follow the guidelines.

For each token in the evaluation set, we label it by majority voting, i.e. the label that two or more annotators agree on. We also aggregate the evaluation data from Maynez et al. (2020) in the same way to produce our own test set for abstract text summarization.

B TRAINING OF NMT MODELS: TRANS2S AND MBART

Models	*TranS2S*	*MBART*
d_{model}	512	1024
d_{hidden}	1024	4096
n_{layers}	6	12
n_{heads}	8	16
p_{dropout}	0.1	0.3

Table 6: Basic hyper-parameters of architecture for NMT models.

Tokenization For TranS2S, we first segment the Chinese corpus with an opensource Chinese word segmentation tool (Luo et al., 2019), then we learn separate BPE vocabularies with 32k merge operations (Sennrich et al., 2016) over the source (Zh) and the tokenized target (En) corpus respectively. For MBART, we directly apply the contained sentence-piece dictionary in the finetuned model to the raw data of Chinese and English corpus.

Model We use the implementation of Transformer model from fairseq (Ott et al., 2019). Following the notations in Vaswani et al. (2017), we show hyperparameters of TranS2S and MBART in Tab. 6.

Training For TranS2S, we apply the standard hyperparameters reported in the example of fairseq. We use the Adam optimizer (Kingma & Ba, 2014) using $\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 1e-8$. The learning rate is scheduled using `inverse_sqrt` with a maximum learning rate 0.0005 and 4000 warmup steps. We set the label smoothing as 0.1. We apply dropout of 0.1 and select the best model with validation BLEU scores. We run the model on 8 GPUs for $300,000$ updates with an effective batch size of around $64,000$ tokens.

When finetuning MBART, we use learning rate of 3e-5, and use `polynomial_decay` for learning rate scheduling with warmup updates of 3,000. The effective batch size is 16,384. Dropout is set to be 0.3 and the attention dropout rate is 0.1. The label smoothing is set to be 0.2. We finetune MBart for 60,000 updates.

Decoding After training, we use beam-search with a fixed beam size 5 for all the models and set the maximum generation length to be twice as the source length.

C Experimental Details for Token-level Hallucination Prediction

Subword Tokenization Depending on the pretrained model (Roberta / XLM-Roberta) we finetune on, we apply corresponding subword segmentation to the synthetic data set (S, T, T') and calculate the edit-distance between the T and T' at the subword level. At evaluation time, the model predicts the hallucination labels for each subword in the sentence, thus we predict a word to be a hallucination word if any subword of it is predicted as a hallucinated one.

Synthetic data generation There are a couple of hyperparameters of noised functions in the BART implementation (Ott et al., 2019). The main noised functions include (1) random masking (randomly replace a word with MASK), (2) random replacement (randomly replace a word with another word in the vocabulary), (3) random insertion of masks. We found that random masking and random replacement are the two key factors affecting the generated sentences and we have provided their settings in the main paper. We apply a random insertion masks rate of 0.2 for all settings. In addition, the noise functions are applied to words instead of spans in our setting.

Finetuning For MT, we finetune a large XLM-Roberta (Conneau et al., 2020) released in fairseq (Ott et al., 2019) which is trained on 2.5T of filtered CommonCrawl data in 100 languages. For summarization, we finetune a large Roberta (Ott et al., 2019) on the synthetic data where we truncate articles that exceed 512 tokens (allowed by the Roberta) to be 512. For both models, we use the Adam optimizer (Kingma & Ba, 2014) with $\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 1e-6$ and weight decay of 0.1. We set the masking probability to be 0.35 for the $\mathcal{L}_{mlm}$ loss. The dropout and attention dropout rates are set to be 0.1. We adopt `polynomial_decay` for learning rate scheduling with learning rate of 2e-5.

D Analysis

Models	Patent		COVID-19	
	True hal. toks (%)	**Pred hal. toks (%)**	**True hal. toks (%)**	**Pred hal. toks (%)**
TranS2S	22.78	22.83	9.44	11.79
MBART	8.84	6.94	6.39	5.55

Table 7: Comparisons of TranS2S and MBART on Patent and COVID-19 domains. **True hal. toks** and **Pred hal. toks** are the ground-truth and our model predicted percentage of hallucinated tokens in the benchmark test set for each domain.

Hallucination Statistics on Patent and COVID-19 We present the percentage of hallucinations from the annotations and our model predictions for the Patent and COVID-10 evaluation data respectively in Tab. 7. We see that both the TranS2S and MBART models produce less hallucinations for the COVID-19 domain although it is a complete out-of-domain test set that has not seen in the training data. We suppose that this is because sentences in COVID-19 are more alike the training domains in terms of writing styles which bring positive transfer. We also note that the predictions from our model match pretty well with human annotations for both of the domains.

Examples In Tab. 8, we present some examples of annotations and our model predictions. We can see that our model performs well in general but can be inaccurate in case of spelling errors of the translations. Besides, we also find some annotation errors while our model predicts correctly.

Reference	the arrangement pattern of the projections 2 will now be explained with reference to figs. 5-7.
Annotation	next,[0] we[0] use[0] fig.[0] 5[0] -[0] 7[0] to[0] explain[0] the[0] disposition[0] pattern[0] with[0] pm-2.[1]
Prediction	next,[0] we[0] use[0] fig.[0] 5[0] -[0] 7[0] to[0] explain[0] the[0] disposition[0] pattern[0] with[1] pm-2.[1]
Reference	a swivel joint 557 is provided in a radially outer region, on an end surface of the drive plate 556.
Annotation	a[0] rotation[0] hinged[1] 557[0] is[0] provided[0] to[0] the[0] external[0] area[0] on[0] a[0] trail[1] that[0] has[0] a[0] preface[1] state.[1]
Prediction	a[0] rotation[0] hinged[0] 557[0] is[0] provided[1] to[0] the[0] external[0] area[0] on[0] a[0] trail[1] that[1] has[1] a[0] preface[1] state.[1]
Reference	r is small, the kinetic energy e of the droplet is small as compared to the air resistance.
Annotation	on[0] the[0] other[0] hand,[0] radius[0] r,[0] which[0] is[0] shorter[1] than[1] the[0] hour[1] in[0] which[1] it[1] can[1] be[1] used,[1] is[0] smaller[0] than[0] the[0] air[0] resistance.[0]
Prediction	on[0] the[0] other[0] hand,[0] radius[0] r,[0] which[0] is[0] shorter[1] than[0] the[0] hour[0] in[1] which[1] it[1] can[1] be[1] used,[1] is[0] smaller[0] than[0] the[0] air[0] resistance.[0]
Reference	if you have a fever of a hundred and two or higher.
Annotation	if[0] your[0] heat[0] reaches[0] 102.d[0] egree.[0] f.[0] or[0] above,[0]
Prediction	if[0] your[0] heat[0] reaches[0] 102.d[1] egree.[1] f.[1] or[0] above,[0]

Table 8: Examples of annotations and model predictions. [0] indicates faithful word while [1] indicates hallucinated word.