

# The Curious Case of Hallucinations in Neural Machine Translation

Vikas Raunak    Arul Menezes    Marcin Junczys-Dowmunt  
Microsoft, USA

{viraunak, arulm, marcinjd}@microsoft.com

## Abstract

In this work, we study hallucinations in Neural Machine Translation (NMT), which lie at an extreme end on the spectrum of NMT pathologies. Firstly, we connect the phenomenon of hallucinations under source perturbation to the Long-Tail theory of Feldman (2020), and present an empirically validated hypothesis that explains hallucinations under source perturbation. Secondly, we consider hallucinations under corpus-level noise (without any source perturbation) and demonstrate that two prominent types of natural hallucinations (detached and oscillatory outputs) could be generated and explained through specific corpus-level noise patterns. Finally, we elucidate the phenomenon of hallucination amplification in popular data-generation processes such as Backtranslation and sequence-level Knowledge Distillation. We have released the datasets and code to replicate our results at <https://github.com/vyraun/hallucinations>.

## 1 Introduction

Neural Machine Translation (NMT) enjoys tremendous success, far surpassing the performance of previous statistical approaches in high-to-moderate resource settings (Koehn and Knowles, 2017). However, NMT suffers from well known pathologies such as coverage (Tu et al., 2016), mistranslation of named entities (Ugawa et al., 2018), etc. In terms of adequacy of the generated output (Martindale et al., 2019), hallucinations are egregious mistakes that lie at the extreme end of NMT pathologies. Such hallucinated outputs are characterized as being decoupled from the source sequence, despite being (fully or moderately) fluent in the target language (Müller et al., 2020). Two main hallucination phenomena have been reported in the existing literature:

1. NMT models tend to generate hallucinated

outputs under certain cases of source perturbation (Lee et al., 2018).

2. NMT models have a propensity to hallucinate more frequently under out-of-domain inputs (Müller et al., 2020).

However, a plausible theory to explain the generation of different types of hallucinations, including the above two results is still lacking in the NMT literature. Lee et al. (2018) posited that hallucinations could be happening due to decoder instability, however, their experiments to engineer solutions based on this proved inconclusive. In this work, we present a systematic study of different kinds of hallucinations, studying them through the lens of generalization, memorization and optimization in sequence to sequence models. Our key contributions are as follows:

1. We extend the Memorization Value Estimator proposed in Feldman and Zhang (2020) to the sequence to sequence setting and demonstrate that hallucinations under source-side perturbations could be explained through the long-tail theory they propose.
2. We introduce corpus-level noise into NMT parallel corpora and show that specific noise patterns interact with sequence to sequence training dynamics in different ways to generate the prominent hallucination patterns reported in the literature (Lee et al., 2018).
3. We demonstrate the phenomenon of hallucination amplification in the outputs generated using Backtranslation (Edunov et al., 2018) and Knowledge Distillation (Kim and Rush, 2016), two widely used data generation algorithms for MT.

## 2 Related Work

Our work connects hallucinations in NMT to the problem of generalization in Deep Learning. In

this section, we briefly survey the two areas.

## 2.1 Hallucinations in NMT

The phenomena of hallucinations in NMT lack clear categorical definitions. Lee et al. (2018) define hallucinations as the model producing a vastly different (inadequate) output when the source is *perturbed* under a specific noise model and present an algorithm to detect such cases. Subsequently, approaches to making NMT models more robust to small perturbations in the input have been actively explored (Cheng et al., 2019), however, no coherent theory to explain the phenomena of hallucinations has been empirically validated in the existing literature. Our work differs from Lee et al. (2018) in that we not only study hallucinations under source side perturbations but also under corpus-level noise. Further, we build on their work by filling in the gap for a plausible hypothesis that explains various types of hallucinations.

Wang and Sennrich (2020) consider hallucinations as outputs detached from the source, and demonstrate that NMT models are more prone to hallucinations under out-of-domain settings by manually ascertaining whether an output generated is hallucinated or not. Manual detection of hallucinations, however, is an impediment for fast experimental cycles, and in this work, besides explaining the generation of such *natural hallucinations* (i.e. hallucinations generated without any source perturbation), we also propose an approximate corpus level hallucination detection algorithm to aid faster analysis.

## 2.2 Generalization in Deep Learning

Feldman (2020) studies label memorization in deep learning, and explains how memorization could be essential for achieving close-to-optimal generalization when the data distribution is long-tailed; since memorizing a representative of a rare subpopulation from the long-tail could significantly increase the prediction accuracy on its subpopulation, thereby improving the generalization error. Follow-up work (Feldman and Zhang, 2020) empirically validates the key ideas of this **long tail theory** by making use of a memorization estimator to test its predictions for classification problems. To the best of our knowledge, our work presents the first study that connects Feldman’s long-tail theory to the problem of hallucinations in NMT.

## 3 Categorizing Hallucinations in NMT

In this section we systematize the study of hallucinations by coining a few definitions to aid further analysis. Firstly, we categorize hallucinations in NMT into two primary categories:

1. Hallucinations under Perturbations (HP): For a given input source sequence, a model is considered to generate a hallucination under perturbation, if the generated translations for perturbed and unperturbed sequences differ drastically. More precisely, we refer to the algorithm proposed by Lee et al. (2018) for detecting hallucinations under perturbation.
2. Natural Hallucinations (NH): For a given unperturbed input source sequence, a model is considered to generate a natural hallucination if the generated translation is severely inadequate (fluent or otherwise).

<p><b>Source:</b> das kann man nur feststellen , wenn die kontrollen mit einer großen intensität durchgeführt werden .</p> <p><b>Correct Translation:</b> this can only be detected if controls undertaken are more rigorous .</p> <p><b>Output:</b> blood alone moves the wheel of history , i say to you and you will understand , it is a privilege to fight .</p>
---

Figure 1: Detached Natural Hallucination Example

<p><b>Source:</b> 1995 das produktionsvolumen von 30 millionen pizzen wird erreicht .</p> <p><b>Correct Translation:</b> 1995 the production reached 30 million pizzas .</p> <p><b>Output:</b> the us , for example , has been in the past two decades , but has been in the same position as the us , and has been in the united states .</p>
--

Figure 2: Oscillatory Natural Hallucination Example: Decoupled from Source + Repeating N-gram Structure

Further, we classify a Natural Hallucination (NH) as belonging to one of the two types:

1. Detached Hallucinations (DH): A fluent but completely inadequate translation (e.g. Figure 1).
2. Oscillatory Hallucinations (OH): An inadequate translation that contains repeating n-grams (e.g. Figure 2).

Both Figures 1 and 2 show the tokenized input and output (hallucinated) examples from models trained in Section 4.2, to illustrate the above two definitions. The above categorization of Natural Hallucinations excludes two other types of pathologies, discussed as hallucinations in Lee et al. (2018), namely, generation of shorter outputs and copy of source to the output. The proposed categorization allows us to quantitatively disentangle the study of hallucinations from other NMT pathologies, without losing any generality.

## 4 Origins of Hallucinations

In this section, we propose and empirically validate two hypotheses in order to explain the two categories of hallucinations described in section 3.

### 4.1 Hallucinations under Perturbations

**Hypothesis 1 (H1)** *The samples memorized by a NMT model are most likely to generate hallucinations when perturbed.*

To validate H1, we adapt the Memorization Value Estimator (MVE) proposed by Feldman and Zhang (2020) to the sequence to sequence setting, by replacing the accuracy metric they use with a sequence overlap metric such as chrF (Popović, 2015) or BLEU (Papineni et al., 2002)<sup>1</sup>. We then compare the hallucination behaviour under perturbation of the most-memorized samples with random samples using the hallucination detection algorithm proposed in Lee et al. (2018).

**Memorization Value Estimation** The modified Memorization Value Estimator (MVE) is described in algorithm 1. MVE computes the memorization value of a sample as the change in average prediction metric  $M$  (for which we use metrics such as chrF, BLEU) for the given sample between the models trained with the sample included in the training set and the models trained with the sample excluded.

**Hallucination Detection** The HP detection algorithm used is presented as algorithm 2. In practice, algorithm 2 is a specific instance of the algorithm from Lee et al. (2018), wherein we make the following three changes:

<sup>1</sup>In practice, other MT metrics such as METEOR or BERT-Score (Banerjee and Lavie, 2005; Zhang et al., 2019) could also be used as empirical extensions of MVE for sequences, however, word/character n-gram overlap provides a stronger indication of memorization than soft-overlap methods like BERT-Score.

---

#### Algorithm 1: Memorization Value Estimator

---

**Data:** Training Dataset  $S$  of size  $n$ , Learning Algorithm  $A$ , Number of Trials  $t$ , Metric  $M$   
**Result:** Memorization Values over  $S$   
 Sample  $t$  random subsets of  $S$  of size  $m$ ;  
**for**  $k = 1$  to  $t$  **do**  
   Train model  $h_k$  by running  $A$  on  $S_k$   
**for**  $i = 1$  to  $n$  **do**  
    $\text{mem}(A, S, i) = \frac{E[M]_{h \leftarrow A(S)}}{E[M]_{h \leftarrow A(S \setminus i)}}$

---

1. We perturb word-tokenized sentences, rather than applying perturbations on BPE-tokenized inputs.
2. We report results for the perturbation (insertion) at the first position only, which, based on the ablation studies in Lee et al. (2018), is the most reliable way to generate hallucinations.
3. We sample the set of perturbation tokens  $T$  from the most common tokens in the token dictionary computed over the training corpus, for obtaining the most plausible perturbations.

---

#### Algorithm 2: Hallucination under Perturbation

---

**Data:** NMT Model, Parallel Corpus ( $X, Y$ ), Token Set  $T$   
**Result:** Hallucinated Samples  $H$   
**for**  $x, y$  in  $X, Y$  **do**  
    $y' = \text{Model}(x)$   
   **if**  $\text{adjusted-bleu}(y', y) > 0.09$  **then**  
     **for**  $t$  in  $T$  **do**  
        $\tilde{x} = \text{put } t \text{ at the beginning of the input } x$   
        $\tilde{y} = \text{Model}(\tilde{x})$   
       **if**  $\text{adjusted-bleu}(\tilde{y}, y') < 0.01$  **then**  
         add  $x$  to  $H$

---

### 4.1.1 Experiments and Results

To compute the memorization values,  $mem$  in algorithm 1, we train  $t = 10$  NMT models using fairseq (Ott et al., 2019) on different randomly selected subsets of sentence pairs (each about 101K samples) from the IWSLT-2014 De-En dataset (160K samples). BPE (Sennrich et al., 2016) with a joint token vocabulary of 10K is applied over lower-cased tokenized text. The NMT model is a six-layer Transformer model with embedding size 512, FFN layer dimension 1024 and 4 attention heads (42M parameters), and the checkpoint with the best validation BLEU (detokenized, with beam=5) is selected. In each case, a batch size of 4K tokens, dropout of 0.3 and tied encoder-decoder embeddings is used. Then, the MVE (algorithm 1) is applied on the training samples using the above  $t$

trained models to compute the memorization values,  $mem$  for each source sample  $i$ . For further analysis, we do not consider any sample which hasn't been excluded from the random training sets at least twice.

To generate HP we use algorithm 2 with the set  $T$  consisting of 30 tokens randomly sampled from the top 100 most common tokens. We apply algorithm 2 to two sets of training samples – a Memorized set comprising of training samples with the highest hundred (100) memorization values, and a Random set (of the same size) sampled from the rest of the training samples. Since, each input sentence can appear in the Hallucinated Samples set  $H$  multiple times in algorithm 2, we report both Unique and Total number of Hallucinations (HP) generated.

We report results using chrF, BLEU as well as the prediction accuracy computed by matching the entire output string to the reference, as the metric  $M$  used in computing the memorization values. Table 1 shows that the difference between the counts of unique HP between the Memorized and Random set is very high. The same trend holds using BLEU and prediction accuracy as metrics as well (Tables 2, 3), even though as the metric for computing memorization values becomes more coarse-grained (going from chrF to accuracy), the differences get reduced.

Set	Unique HP	Total HP
<b>Random</b>	1	1
<b>Memorized</b>	51	431

Table 1: Memorized vs Random Set Comparison using Algorithm 2 with chrF as the Metric in Algorithm 1.

Set	Unique HP	Total HP
<b>Random</b>	2	2
<b>Memorized</b>	42	180

Table 2: Memorized vs Random Set Comparison using Algorithm 2 with BLEU as the Metric in Algorithm 1.

Set	Unique HP	Total HP
<b>Random</b>	4	5
<b>Memorized</b>	8	30

Table 3: Memorized vs Random Sets Comparison using Algorithm 2 with Accuracy as the Metric in Algorithm 1.

**Further Comparisons** Figure 3 (Top) presents the number of unique hallucinations (using BLEU

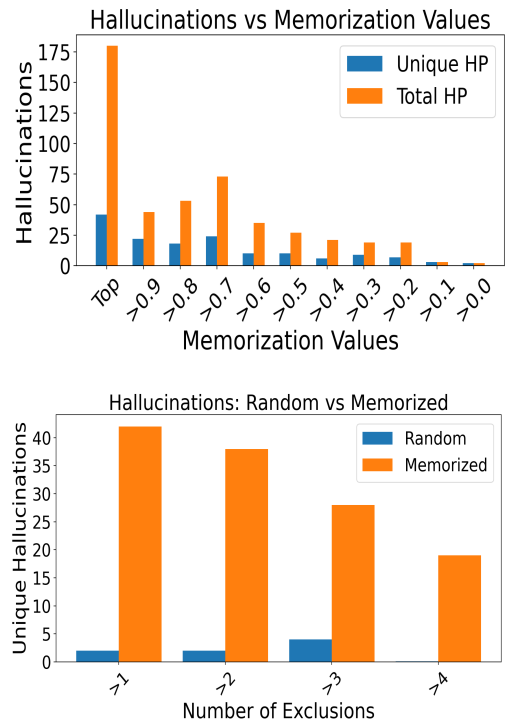


Figure 3: Further Comparisons: (Top) measures Hallucinations under increasingly restrictive sampling sets, in terms of the memorization value. (Bottom) compares the Memorized vs Random sets under different number of sample exclusions

as the metric in algorithm 1, as in Table 2; the default metric from hereon, unless stated otherwise), when the underlying sampling set for constructing the set under evaluation, is restricted using different threshold memorization values (varying from 0 to 0.9, in increments of 0.1). The figure shows that as the memorization values increase, the number of unique (Unique HP) as well as total hallucinations (Total HP) keeps increasing as well, demonstrating a strong positive correlation between hallucination frequency and memorization values.

Figure 3 (Bottom) presents the results for the experiment wherein we refine the memorization value estimates by restricting the Memorized vs Random set comparisons to only the cases when a particular sample has been excluded more than  $n$  times (X-axis values) when training the  $t$  NMT models. Here, we find that the trend of large differences between the counts of unique hallucinations generated for the two sets stays consistent as the memorization value estimates are made more accurate. In fact, when the two sets (Random, Memorized) are constructed only over the samples which have been excluded at least 4 times, we find zero unique HP for the Random set.

**Encoder-Decoder Attention Analysis** To further analyze how memorized samples suffer more hallucinations under perturbations, we compare the cross-attention heads of the last layer of the decoder for the Random and Memorized sets. Table 4 presents a comparison of the average entropy of the attention matrix, averaged diagonal attention and the average attention paid to the last source token, aggregated over the entire sets. The results show that the two sets differ considerably in terms of the attention distribution, with the memorized set having more fixed (lower-entropy) average attention distributions. Although this result is known for hallucinated translations (Lee et al., 2018; Voita et al., 2020; Berard et al., 2019), which have a tendency of producing deficient attention maps, the fact that this phenomenon extends to memorized samples as well further helps establish the link between memorization and hallucination under perturbation.

Data-Type	Memorized	Random
Attention Entropy	<b>1.85</b>	<b>2.70</b>
Diagonal Attention Entropy	1.48	2.21
Average Last Token Attention	<b>0.35</b>	<b>0.25</b>

Table 4: Attention Statistics Comparison for Random vs Memorized Sets.

## 4.2 Natural Hallucinations

**Hypothesis 2 (H2)** *Corpus-level noise patterns (comprised of invalid source-target pairs) dictate the type of natural hallucinations generated by the NMT model.*

Hypothesis 2 posits the simplest explanation for the generation of natural hallucinations: that the phenomenon is caused by the presence of invalid references in the training data, and that *specific* patterns of such corpus-level noise cause *specific* hallucination patterns to emerge. Establishing a causal link between corpus-level noise patterns and hallucination types could greatly ease diagnosing the origins of such cases.

We try to validate H2 by construction: first, we build four different types of the corpus-level noise patterns, and then we analyze the resulting models in terms of the generated translations.

### 4.2.1 Experiments and Results

We train 5 models on the IWSLT 2014 corpus, where the training data consists of 160K samples. We train a baseline model with no noise, while the other 4 models are trained with specific patterns of added noise. The model and training settings are

the same as in section 4.1, except that BPE is now learnt on the noise-added corpus for the 4 models.

**Corpus-Level Noise Model** In order to generate the noise sets to be added to the training parallel data, we first construct an invalid reference set (IRS), a small set of detached source-target pairs and use the larger WMT 2014 De-En corpus as an additional data source (the size of the constructed IRS is 21 for the below experiments). Then, the different noise sets (of the same size) are constructed using different sampling strategies for sources and targets, which combine source-target sequences drawn from the IRS and the WMT 2014 De-En training corpus into noise sets with particular characteristics. Specifically, we generate the noise sets as follows:

1. Unique-Unique (UU): We sample 21K<sup>2</sup> random unique source sentences from WMT, and pair each with an unrelated unique random target sentence from WMT.
2. Repeat-Repeat (RR): We sample 21 unique source sentences from IRS, and pair each with unrelated unique random target sentence from IRS, and repeat each such pair 1000 times.
3. Repeat-Unique (RU): We use the same 21 random unique source sentences as RR. We repeat each 1000 times, and pair each repeat with unrelated unique random target sentence from WMT.
4. Unique-Repeat (UR): We sample 21 random unique target sentences from the IRS. Each such target sentence is repeated 1000 times. Each repeat is paired with an unrelated unique random source sentence from WMT.

**Evaluation** We train NMT models with each of the above four noise sets added to the IWSLT De-En parallel corpus, and report the results for both De-En and En-De translation directions. Specifically, we investigate the behavior of models trained on each of the above noise sets using the following evaluation sets:

1. IWSLT: The IWSLT De-En 2014 test set, which does not overlap with the training data, is used to measure generalization.

<sup>2</sup>21K amounts to approximately 12% noisy samples, when combined with the 160K parallel training samples for the IWSLT De-En corpus.

2. Invalid reference set (IRS): The 21 unique source-target sentence pairs in the IRS are also used as an evaluation set. Due to the way the noise sets are built, the IRS overlaps with the various training sets: it is contained in the RR training data, its source sentences are present in the RU training data and its target sentences are present in the UR training data, while there is no overlap for the UU training data. The main purpose of evaluating models on this set is to measure memorization of the overlapping source/targets.
3. Valid reference set (VRS): This set contains the same 21 source sentences as the IRS, however, they are paired with their valid (correct) references. The VRS set is used to measure whether the NMT model can generalize despite the presence of source/targets associated with the noise sets.

Using the above evaluation sets, we then compute the following metrics:

- BLEU: BLEU score for each evaluation set.
- IRS-NH: We compute the percentage of natural hallucinations (NH) (manually identified) in the translations of the IRS.
- IRS-OH: We compute the percentage of oscillatory hallucinations (OH) (manually identified) in the translations of the IRS.
- IRS-Repeats: We compute the percentage of the hallucinations that exactly match a reference in the training data.
- IRS-Unique Bigrams: We compute the number of unique bigrams in the translations of the IRS, as a fraction of total possible unique bigrams in sentences of the same length.

**Design of Noise patterns** While the above noise patterns are quite plausible in a web-based corpus collection process, due to the widespread adoption of automatic bitext mining algorithms (Schwenk, 2018) applied over noisy sources, our primary motivation behind constructing these four types of noise patterns is to present different optimization scenarios for the NMT model under training. In each of the four noise patterns, the source-target pairs are ‘invalid’, but the difference lies in the number of representation pathways (contexts) each

**Source:** das ist eine unerfreuliche situation , die wir künftig vermeiden wollen .  
**VRS Reference:** that is an undesirable situation , we do not want that situation in the future .  
**No Noise Output:** this is an unpleasant situation that we &apos;re trying to avoid in the future .  
**UU Output:** *the us , in particular , is not alone .*  
**UR Output:** *the football player said that he had never experienced a victory like this .*  
**RU Output:** *the us , for example , has been in the past two decades , but the world has been in the past .*  
**RR Output:** *that is what she said .*

Figure 4: Sample Outputs under Corpus-level Noise

set offers for the ‘invalid error’ to propagate to the different layers, imposing a different set of requirements on the underlying optimization process. We posit that the four different noise patterns (RU, UR, UU, RR) interact in different ways with the encoder and decoder of an NMT model, e.g. for the RU noise pattern, the decoder is required to generate unique translations for the same sources, thereby encouraging decoder instability, whereas under the UR noise pattern, the encoder is required to produce the same representations for unique inputs, allowing the ‘invalid error’ to propagate to lower encoder layers. In UU noise as well, the model is required to produce encoder representations that are vastly different in the representation similarity space (when compared to the rest of the training corpus), while offering multiple contexts for the invalid error to propagate, while in the case of RR noise, the invalid error propagation is quite restricted. Further, we can test whether the above hypotheses have any predictive power through the properties of the generated translations of noisily trained models. However, a rigorous exploration of the impact of noise patterns on encoder-decoder training dynamics is out of scope for this work.

**Results** Tables 5 and 6 show the results for both the De-En and the En-De translation directions. The boxes marked with ‘-’ are the cases where the associated metric computation does not convey any useful information. We see the following patterns in the results:

1. The Test-BLEU is not greatly affected by the noise, except in the UR case, with the models matching the baseline (trained with no noise).

Noise	IRS-BLEU	VRS-BLEU	Test-BLEU	IRS NH	IRS OH	IRS Repeats	IRS Unique-Bigrams
<b>U-U</b>	0.48	12.90	33.75	33.33 %	0.0 %	0.0 %	0.80
<b>U-R</b>	12.20	12.82	33.26	42.85 %	4.76 %	38.09 %	0.84
<b>R-U</b>	0.42	0.55	34.03	-	76.19 %	0.0 %	0.19
<b>R-R</b>	100.00	0.39	33.89	-	-	-	0.96
<b>None</b>	0.54	20.59	33.65	0.0 %	0.0 %	0.0%	0.925

Table 5: Analysis of Models trained using different Corpus-level Noise Patterns: De-En

Noise	IRS-BLEU	VRS-BLEU	Test-BLEU	IRS NH	IRS OH	IRS Repeats	IRS Unique-Bigrams
<b>U-U</b>	0.25	13.26	28.61	14.28 %	0.0 %	0.0 %	0.91
<b>U-R</b>	0.42	12.15	27.93	9.52 %	0.0 %	9.52 %	0.95
<b>R-U</b>	0.35	0.56	28.67	-	47.61 %	0.0 %	0.49
<b>R-R</b>	100.00	0.50	28.54	-	-	-	0.99
<b>None</b>	2.06	14.62	28.33	0.0 %	0.0 %	0.0%	0.95

Table 6: Analysis of Models trained using different Corpus-level Noise Patterns: En-De

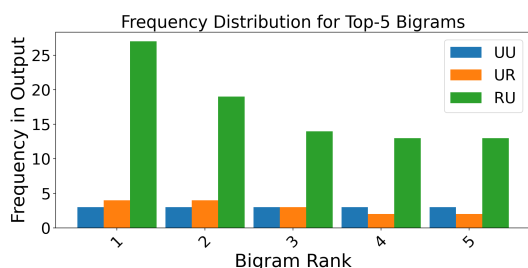


Figure 5: Frequency for the Top 5 Bigrams in the Output for the IRS for the different noisy models on IWSLT 2014 En-De (Table 6).

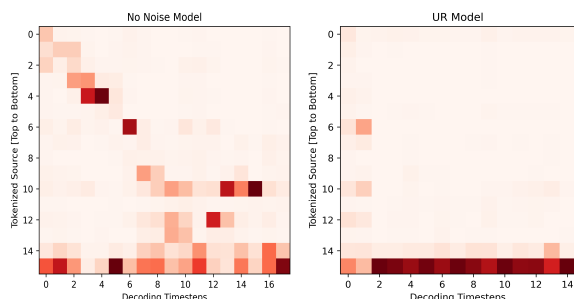


Figure 6: Attention Visualization for translation of source in Figure 4, on which the UR model hallucinates (right), compared against the model with no noise (left). The right attention map displays the characteristic hallucination pattern (Lee et al., 2018; Berard et al., 2019; Voita et al., 2020). The source is not present in the training corpus for the UR model. However, source sequences from the same domain (WMT corpus) are present with invalid references.

2. When we consider the IRS-BLEU, we find that the RR model has fully memorized this data. This is to be expected as it has seen this set repeated 1000 times.
3. On the IRS set, the UR model produces a num-

ber of repeated outputs (IRS Repeats) from the training corpus.

4. On the IRS set, the RU model produces a very high percentage of oscillatory hallucinations (OH).

#### Linking Hallucination Patterns to Noise Patterns

The main purpose of the above experiments is to demonstrate how natural hallucinations can be generated on source sequences seen or unseen during training, and their relation to specific noise types. The link between noise patterns and specific types of hallucinations in the output could be used as very effective diagnostic tool to trace hallucinated outputs to corpus-level noise, with the goal of removing the noise from the training dataset.

In this regard, two important observations further emerge from Tables 5 and 6. First, that in the case of UR noise, a considerable percentage of natural hallucinations (IRS NH) manifests as a direct copy of a training reference (without any of the IRS source sequences being present in the training set). Second, for the case of RU noise, oscillatory hallucinations (OH) are very prominent, as evident by the number IRS Unique-Bigrams, which are considerably lower when compared to the other noise types. Figure 5 presents the comparisons for counts of the top 5 bigrams present in the translations of the IRS set, showing how among the 4 noise patterns, RU leads to the most oscillatory hallucinations. Resulting sets of translations for a source sequence present in the IRS is shown in Figure 4, while Figure 6 presents a qualitative comparison of the attention patterns for this source sequence.

## 5 Hallucination Amplification

In this section, we analyze how hallucinations caused due to corpus-level noise get amplified when a model trained on a noisy MT corpus is used for downstream data generation in algorithms such as Sequence-level Knowledge Distillation (KD) (Kim and Rush, 2016) and Backtranslation (BT) (Edunov et al., 2018). To analyze this, we need to compute NH at scale. So, firstly, we propose an automatic NH detection algorithm based on the analysis that hallucinations often occur in terms of oscillations or repeats of the target sequences.

---

### Algorithm 3: Corpus-level NH Estimator

---

**Data:** Source  $S$ , Multi-lingual Similarity Scoring Model  $X$ , NMT Model  $M$ , Noise Estimate  $\epsilon$ , N-gram order  $n$ , Threshold  $t$

**Result:** Approximate Natural Hallucinations ANH

$T$  = Decode the Source Sequences in  $S$ ;

$S_X$  = Compute Cross-Lingual Similarity for  $(T, S)$  ;

$F_1$  = Select Translations where the count of the top repeated  $n$ -gram in the translation is greater than the count of top repeated source  $n$ -gram by at least  $t$  ;

$F_2$  = Select Translations in  $T$  that are Paired with Multiple Unique Sources ;

$S_\epsilon$  = Bottom  $\epsilon$  percentage of samples in  $S_X$  ;

$ANH = (S_\epsilon \cap F_1) \cup (S_\epsilon \cap F_2)$

---

The proposed NH Estimator (algorithm 3) is reference-free and works at the corpus-level. One simplifying assumption used in algorithm 3 is that the repeats are now computed on the translations generated over the source set rather than on the training set (as in Tables 5 and 6 for the IRS-Repeats metric). The motivation behind this assumption is that given a sufficiently large source set, the translated output (if hallucinated as a direct copy of one of the training set targets), will appear more than once in the decoded set (since UR noise is one of its causes).

### 5.1 Experiments and Results

We use algorithm 3 to measure NH caused by using the models trained on the noisy corpora (as explored in section 4.2 and analyzed in Tables 5 and 6) for BT and Sequence-level KD. For BT, we use 1 million English sentences from the WMT 17 De-En dataset as the monolingual corpus and generate back-translations via sampling (Edunov et al., 2018), using the different types of noisily trained models (RR, UU, UR, RU) for En-De. For constructing a sequence-level KD dataset we generate the translations over the initial IWSLT 2014 De-En corpus training corpus (the initial parallel

data, with no noise) with a beam size of 5 (Kim and Rush, 2016). The results of applying the NH estimator (with  $\epsilon = 1$ , i.e. 1%,  $n = 4$ ,  $t = 2$  and LASER as the cross-lingual similarity scoring Model  $M$  (Artetxe and Schwenk, 2019)) on the outputs generated using KD and BT are presented in Table 7 and Table 8 respectively.

Data-Type	$F_1$	$F_2$	$S_\epsilon \cap F_1$	$S_\epsilon \cap F_2$	$ANH$
Parallel	17	1900	0	41	41
KD-None	17	2310	0	120	120
KD-UU	17	2394	1	118	119
KD-UR	30	<b>3090</b>	0	688	688
KD-RU	24	2365	2	122	124
KD-RR	29	2411	1	135	136

Table 7: Hallucination Amplification for Knowledge Distillation (Kim and Rush, 2016)

Data-Type	$F_1$	$F_2$	$S_\epsilon \cap F_1$	$S_\epsilon \cap F_2$	$ANH$
BT-UU	154	7592	0	5009	5009
BT-UR	43	<b>176184</b>	0	7280	7280
BT-RU	56	67	1	2	3
BT-RR	62	65	1	11	12

Table 8: Hallucination Amplification for Backtranslation (Edunov et al., 2018)

We find that the UR models lead to severe amplifications for both BT and KD. For KD, we find that all noisy models lead to increase in NH when compared to the initial parallel corpus (implying amplification), which itself contains a non-trivial number of repeated targets. For BT, both UU and UR models lead to large number of repeated generations. RR models however cause the least hallucinations for both KD and BT. Our proposed NH estimator is not able to detect many OH however, in any of the cases due to very little overlap with the bottom  $\epsilon = 1\%$  similarity scores, even though the  $F_1$  column indicates amplification of translations with repeated n-gram patterns ( $F_1$ ) in the KD datasets.

Further, since, there is hallucination amplification going from a parallel corpus to the KD data generated (using noisy models trained on the parallel corpus), downstream systems trained on the KD data will be impacted in terms of hallucinations as well. We leave further downstream analysis to future work.

## 6 Discussion

In this section, we present a qualitative analysis of a few topics discussed in section 4, along with a discussion on some future research directions.



MV	Source	Target
0.61	gerade plus gerade : gerade . ungerade plus ungerade : gerade .	even plus even gives you even . odd plus odd gives you even .
0.65	also ist $2 \wedge 5 : 2 \times 2 = 4, 8, 16, 32$ .	so $2 \wedge 5$ is $2 \times 2 = 4, 8, 16, 32$ .
0.69	beweg dich ! nein ! beweg dich ! nein ! beweg dich ! nein !	move . no . move . no . move . no .
0.82	frau : sie bestanden darauf , ich würde lügen .	they insisted that i was lying .
0.94	mjam , mjam , mjam , mjam , mjam .	gobble , gobble , gobble , gobble , gobble .

Table 9: Examples of Samples from the Top-100 Most Memorized Samples in the Training Set as measured using the Memorization Value (MV) Estimator (Algorithm 1) with chrF as the Metric: De-En.

## 6.1 Memorized Samples

Table 9 presents some examples from the most memorized training samples, thereby representing the samples from the long-tail of the data that is likely to have been memorized by the model. Qualitatively, the examples appear to be different (in terms of source/target syntax) from a random subset of training samples (e.g. in Appendix A, Table 10), although we leave further quantitative analysis of the differences to future work. Similarly, the link between out-of-domain and memorized samples needs to be ascertained quantitatively.

## 6.2 Preventing Hallucinations

In this subsection, we discuss a few methods that could be effective in preventing hallucinations.

**Data-Augmentation** To prevent hallucinations under perturbation resulting from memorization of the samples in the long-tail of the dataset (Feldman, 2020), a simple iterative solution could be to analyze the long-tail (using Algorithm 1), and implement data-augmentations specific to the characteristics of such samples (e.g. as in Table 9), with the goal of bringing such samples out of the long-tail (Raunak et al., 2020). Further work is required to determine the dynamics of such transition.

### Ameliorating Memorization During Learning

Robust learning algorithms e.g. Robust Early learning (Xia et al., 2021) that are designed to prevent memorization specifically are likely to prevent perturbation based hallucinations.

### Robust Learning on Noisy Samples

Kang and Hashimoto (2020) propose a loss-truncation approach to reduce the impact of noisy references in sequence-to-sequence training, using the intermediate model’s loss as a sample quality estimator and test their algorithm on a summarization task. Li et al. (2021) present a modification to Expected Risk Minimization (ERM), namely Tilted-ERM to reduce the impact of outliers during training. Such techniques could be useful in increasing learning robustness to corpus-level noise in NMT as well.

**Corpus-Level Filtering** Incorporating heuristics or filters (Junczys-Dowmunt, 2018; Zhang et al., 2020) to remove invalid source-target pairs, especially the noise patterns explored in section 4.2 (or to remove bitext indeterminacy in general) could be effective in reducing natural hallucinations.

## 7 Conclusion

In this work we demonstrated that memorized training samples are far more likely to hallucinate under perturbation than non-memorized samples, under an extension of the Memory Value Estimator proposed in Feldman and Zhang (2020). We also showed that specific noise patterns in the training corpora lead to specific well-known hallucination patterns. Finally, we demonstrated that these patterns can be amplified by popular data-generation processes such as backtranslation and sequence-level knowledge distillation.

Due to the compute-intensive algorithms involved in our analysis, we conduct most of our experiments using the IWSLT 2014 corpus. However, long-tailed phenomena are a characteristic of natural language and even scaling the size of the corpus doesn’t alleviate the characteristic Zipfian distribution of the occurrence of words/tokens in the NMT corpora; which, according to the central thesis of the long-tail theory (Feldman, 2020), would lead to memorizations. Similarly, noise in the form of invalid references is an artifact of the scale at which web-based corpora are collected and given that both hallucinations under perturbations and natural hallucinations are widely reported in large-scale NMT systems, our insights should be directly applicable to larger-scale models as well.

We hope that our work serves as a useful step towards a detailed understanding of hallucinations in NMT and in other sequence to sequence models. Among the numerous interesting directions for follow-up work, in future, we would like to explore learning-centric fixes to ameliorate the impact of memorization and corpus-level noise patterns in NMT training.

## References

- Mikel Artetxe and Holger Schwenk. 2019. [Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond](#). *Transactions of the Association for Computational Linguistics*, 7:597–610.
- Satanjeev Banerjee and Alon Lavie. 2005. [METEOR: An automatic metric for MT evaluation with improved correlation with human judgments](#). In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.
- Alexandre Berard, Ioan Calapodescu, and Claude Roux. 2019. [Naver labs Europe’s systems for the WMT19 machine translation robustness task](#). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 526–532, Florence, Italy. Association for Computational Linguistics.
- Yong Cheng, Lu Jiang, and Wolfgang Macherey. 2019. [Robust neural machine translation with doubly adversarial inputs](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4324–4333, Florence, Italy. Association for Computational Linguistics.
- Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. [Understanding back-translation at scale](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.
- Vitaly Feldman. 2020. [Does learning require memorization? a short tale about a long tail](#). In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2020, page 954–959, New York, NY, USA. Association for Computing Machinery.
- Vitaly Feldman and Chiyuan Zhang. 2020. [What neural networks memorize and why: Discovering the long tail via influence estimation](#). *Advances in Neural Information Processing Systems*, 33.
- Marcin Junczys-Dowmunt. 2018. [Dual conditional cross-entropy filtering of noisy parallel corpora](#). In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 888–895, Belgium, Brussels. 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.
- Yoon Kim and Alexander M. Rush. 2016. [Sequence-level knowledge distillation](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.
- Philipp Koehn and Rebecca Knowles. 2017. [Six challenges for neural machine translation](#). In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.
- Katherine Lee, Orhan Firat, Ashish Agarwal, Clara Fannjiang, and David Sussillo. 2018. [Hallucinations in neural machine translation](#). In *NIPS 2018 Interpretability and Robustness for Audio, Speech and Language Workshop*.
- Tian Li, Ahmad Beirami, Maziar Sanjabi, and Virginia Smith. 2021. [Tilted empirical risk minimization](#). In *International Conference on Learning Representations*.
- Marianna Martindale, Marine Carpuat, Kevin Duh, and Paul McNamee. 2019. [Identifying fluently inadequate output in neural and statistical machine translation](#). In *Proceedings of Machine Translation Summit XVII Volume 1: Research Track*, pages 233–243, Dublin, Ireland. European Association for Machine Translation.
- Mathias Müller, Annette Rios, and Rico Sennrich. 2020. [Domain robustness in neural machine translation](#). In *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 151–164, Virtual. Association for Machine Translation in the Americas.
- 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 the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. [Bleu: a method for automatic evaluation of machine translation](#). In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
- Maja Popović. 2015. [chrF: character n-gram F-score for automatic MT evaluation](#). In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.
- Vikas Raunak, Siddharth Dalmia, Vivek Gupta, and Florian Metze. 2020. [On long-tailed phenomena in neural machine translation](#). In *Findings of the Association for Computational Linguistics: EMNLP*

2020, pages 3088–3095, Online. Association for Computational Linguistics.

Holger Schwenk. 2018. [Filtering and mining parallel data in a joint multilingual space](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 228–234, Melbourne, Australia. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. [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)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. [Modeling coverage for neural machine translation](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 76–85, Berlin, Germany. Association for Computational Linguistics.

Arata Ugawa, Akihiro Tamura, Takashi Ninomiya, Hiroya Takamura, and Manabu Okumura. 2018. [Neural machine translation incorporating named entity](#). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3240–3250, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Elena Voita, Rico Sennrich, and Ivan Titov. 2020. [Analyzing the source and target contributions to predictions in neural machine translation](#). *arXiv preprint arXiv:2010.10907*.

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.

Xiaobo Xia, Tongliang Liu, Bo Han, Chen Gong, Nannan Wang, Zongyuan Ge, and Yi Chang. 2021. [Robust early-learning: Hindering the memorization of noisy labels](#). In *International Conference on Learning Representations*.

Boliang Zhang, Ajay Nagesh, and Kevin Knight. 2020. [Parallel corpus filtering via pre-trained language models](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8545–8554, Online. Association for Computational Linguistics.

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

## A Appendices

Table 10 presents some random samples from the lower-end of the memorization values. The sentences differ in terms of their syntactic properties (versus Table 9 in section 6.1), although more analysis is required to quantitatively ascertain the differences.

Table 10: Examples of Samples with Memorization Value (MV) between 0.0 and 0.6 (Below Top-100) from the Training Set as measured using the Memorization Value (MV) Estimator (Algorithm 1) with chrF as the Metric: De-En.

MV	Source	Target
0.0	sie haben das gleiche gehirn , und das gleiche eeg .	they have the same brain , and the same eeg .
0.08	und wir hatten keine ahnung , wo er war .	it looks like the technological problem is solved .
0.25	also , die wiederbesiedlung ist wirklich sehr langsam .	so , recolonization is really very slow .
0.30	zum einen waren beide sehr real .	for one thing , both were very real .
0.50	wir überqueren auf alu-leitern mit angefügten sicherheits-seilen .	we cross on aluminum ladders with safety ropes attached .