Hallucinations in neural machine translation

Abstract

Neural machine translation (NMT) systems have reached state of the art performance in translating text and widely deployed. Yet little is understood about how these systems function or break. Here we show that NMT systems are susceptible to producing highly pathological translations that are completely untethered from the source material, which we term hallucinations. Such pathological translations are problematic because they are disturbing of user trust and easy to find. We describe a method to generate hallucinations and show that many common variations of the NMT architecture are susceptible to them. We study a variety of approaches to reduce the frequency of hallucinations, including data augmentation, dynamical systems and regularization techniques and show that data augmentation significantly reduces hallucination frequency. Finally, we analyze networks that produce hallucinations and show signatures of hallucinations in the attention matrix and in the stability measures of the decoder.

1 Introduction

Neural machine translation (NMT) systems are language translation systems based on deep learning architectures (Cho et al., 2014a; Bahdanau et al., 2014; Sutskever et al., 2014). In the past few years NMT has vastly improved and has been deployed in production systems, for example at Google (Wu et al., 2016), Facebook (Gehring et al., 2017), Microsoft (Hassan et al., 2018) and many others. NMT systems exhibit both the strengths and weaknesses of the deep learning systems. For example, NMT systems are competitive with state of the art performance (Bojar et al., 2017) and scale well to very large datasets (Ott et al., 2018) but like most large deep learning systems, NMT systems are poorly understood. For example, repeating a word many times into some commercial translation systems sometimes results in strange translations, a phenomenon which has been highly publicized (Christian, 2018). More broadly, recent work shows that NMT systems are highly sensitive to noise in the input tokens (Belinkov & Bisk, 2017) and are also susceptible to adversarial inputs (Cheng et al., 2018). When there is an error in translation, it can be challenging to either understand why the mistake occurred or engineer a fix.

Here, we study in noise input sequences and describe a type of phenomenon that is particularly pernicious, whereby a single additional input token inserted into the source sequence can completely divorce the translation from the input sentence. For example, here is a German input sentence translated to English (reference) by a small NMT system:

Source: Caldiero sprach mit E! Nachrichten nach dem hart erkämpften Sieg,
noch immer unter dem Schock über den Gewinn des Großen Preises von 1 Million $.

Reference: Caldiero spoke with E! News after the hard-fought victory, still
in shock about winning the $1 million grand prize.

NMT Translation: Caldiero spoke with E, after the hard won victory,
still under the shock of the winning of the Grand Prix of 1 million $.
Simply by adding a single input token (mit: with, werden: to become and dass: that) to the beginning of the input sentence and translating them with the same NMT model yield the following:

Mistranslations:

mit: It was said to have a lot of fun in the world.
werden: Don’t hesitate to contact us, if you want to be able to pay for you.
dass: I’m looking forward to having a lot of money.

These mistranslations are completely semantically incorrect and also grammatically viable. They are untethered from the input so we name them ‘hallucinations’. Clearly, even if hallucinations occur only occasionally, the NMT model will be problematic from the perspective of user trust, potentially leading the user to a false sense of confidence in a very incorrect translation. In this work, we show that hallucinations are widespread in the popular NMT system we study. For example, 73% of sentences in our development set can be perturbed to hallucination in our simplified canonical model. If we decode with beam search that number drops, but remains high at 48%.

We systematically explore hallucinations in a variety of NMT variants. We also develop methodologies to help ameliorate this problem, using ideas from data augmentation, dynamical systems theory, and regularization. Finally, we analyze an NMT model that demonstrates hallucinations and show that there are signatures of hallucinations that can be seen in the attention matrix. We focus on NMT systems built using RNNs (Wu et al., 2016), on which many commercial translation systems are based. As RNNs recursively generate translations, we examine the dynamical nature of RNNs. Finally, we study decoder stability when hallucinations are produced and when they are not.

2 Related work

Since its invention, researchers have been working to better understand NMT. For example, moving from the original Seq2Seq model Sutskever et al. (2014); Cho et al. (2014b) to models that utilize attention mechanisms (Bahdanau et al., 2014), resulted in improved translation quality (Luong et al., 2015) and better interpretability (Ding et al., 2017). Studies identified the most critical components of the LSTM (Greff et al., 2017) and the role of the LSTM in language modeling (Karpathy et al., 2015) more broadly. Followed by explorations in interpretability, recent work has focused on robust NMT, studying the effects of input noise, aiming to reduce variation from typos (Belinkov & Bisk, 2017) and synonym choice (Cheng et al., 2018) in the discrete input set used by NMT systems. Both (Belinkov & Bisk, 2017) and (Cheng et al., 2018) have discovered that NMT systems are highly sensitive to input noise and both used adversarial training to help stabilize NMT systems (either with black-box adversarial training or augmenting an adversarial loss). There has been work in understanding how to handle some of the pathologies in RNNs for language modeling and translation, for example, using scheduled sampling to handle mismatches between training and test sets (Bengio et al., 2015).

In parallel, there has also been work in understanding RNNs through the framework of dynamical systems (Sussillo & Barak, 2013; Sussillo, 2014; Laurent & von Brecht, 2016; Rivkind & Barak, 2017; Beer & Barak, 2018) and understanding their capacity in language tasks (Collins et al., 2017). For example, it is known that continuous time vanilla RNNs exhibit high-dimensional chaos (Sompolinsky et al., 1988), which can be driven out by strong input (Rajan et al., 2010). RNNs exhibiting chaos can be beneficial, for example, RNNs capable of chaos in the absence of input can serve as strong initialization for training RNNs when they are input driven (Sussillo & Abbott, 2009), but caution must be used as unbridled chaos can be viewed as a source of dynamical noise. This has led to efforts to rid RNNs of chaos altogether (Laurent & von Brecht, 2016). Efforts related to improving optimization may also have dynamically regularizing effects, e.g. (Arjovsky et al., 2015). Given the complexities and slow compute times of recurrent systems, there have also been attempts to rid NMT of recurrence (Kalchbrenner et al., 2016; Gehring et al., 2017; Vaswani et al., 2017).

We further expand on these studies by highlighting the specific pathology of hallucinations, systematically studying those hallucinations, and analyzing them from a dynamical systems perspective.
3 NMT Methods

Models: In this paper, we use a standard RNN-based encoder-decoder NMT model with attention. Specifically, we study the NMT model described in (Wu et al., 2016), known as GNMT. We use the GNMT model and its public implementation\footnote{https://github.com/tensorflow/nmt with gnmt-v2 architecture.}. Formally, given an input sequence, \( x_{1:S} \) of length \( S \), the NMT model first encodes the input sequence \( x_{1:S} \) into a set of vectors \( z_{1:S} = f_{\text{enc}}(x_{1:S}) \) using its encoder \( f_{\text{enc}} \). The task of the decoder \( f_{\text{dec}} \) is to generate the translation, \( y_{1:T} \), one symbol at a time \( y_t \), given the encoding, \( z_{1:S} \), and previously generated symbols \( y_{<t} \). The decoder, \( f_{\text{dec}} \) is implemented as a conditional sequence model (Bahdanau et al., 2014), where the distribution over \( y_{1:T} \) is conditioned on \( x_{1:S} \). The decoder internally makes use of an attention mechanism \( f_{\text{att}} \) to query the encoder, summarizing \( z_{1:S} \) for each output symbol \( y_t \), putting it all together \( y_t = f_{\text{dec}}(y_{<t}, f_{\text{att}}(z_{1:S})) \) (also see Figure 6 for a detailed model schematic). Finally, the conditional probability of the target sequence is modelled as \( p(y_{1:T}|x_{1:S}) = \prod_{t=1}^{T} p(y_t|y_{<t}, x_{1:S}) \) and the log of this conditional likelihood is maximized given a set of source-target pairs \((x,y)\) during training.

For the sake of research tractability, we study models that are significantly smaller than those used in state-of-the-art or production systems. We use a single layer bidirectional LSTM in the encoder \( f_{\text{enc}} \) and a two layer unidirectional LSTM in the decoder \( f_{\text{dec}} \) with an additive attention mechanism \( f_{\text{att}} \) (Britz et al., 2017). The dimensions of word embeddings and each LSTM hidden cell (both in the encoder and decoder) are set to 256. We refer to this model as the canonical model. Unless otherwise stated, we used Adam (Kingma & Ba, 2014) with a learning rate of 0.001, a constant learning rate schedule, and clipped gradients to a maximum norm of 5.0 during training.

Given these hyper-parameter and architectural choices, we trained 10 canonical models with different random seeds to observe how parameter initialization variability played a role in our results. All additional model variants were also trained 10 times with the same 10 different random seeds. Each model was trained for 1M steps (updates) with a mini-batch of size 128. We selected the checkpoint with the best BLEU score on the development set.

The central goal of our study was to understand how various modelling choices affected the frequency of hallucinations. In order to isolate the effects of modeling changes, all model variants we study in this paper were identical to the canonical model except for a single change. This means, for example, that our model with 512 hidden units also is 2 layers deep, etc. We performed a simple hyper-parameter search for the canonical model, and did not perform additional hyper-parameter searches for any additional models. All models we present are well trained with a BLEU score of at least 20.0 on the test set, a reasonable score for 2-layer models with 256 hidden units.

Inference: Generating a translation of the input sequence, or formally, finding an output sequence that maximizes the conditional log-probability, \( \hat{y} = \arg\max_x \log p(y|x) \), is a major challenge in NMT since the exact inference (or decoding) is intractable. NMT uses approximate decoding techniques, the simplest of which, greedy decoding, chooses the most-likely symbol under the conditional probability \( \hat{y}_t = \arg\max_{i} \log p(y_t = i|y_{<t}, x_{1:S}) \), outputting a single best local prediction by keeping track of a single hypothesis \( k \), at each time step. Another approximate decoding technique, \( \text{beam search} \), improves on greedy decoding by keeping track of multiple hypotheses (\( \text{beams} \)), where \( k > 1 \) at each time step of the decoding, compared to \( k=1 \) in greedy-decoding. To maintain simplicity, we used greedy decoding in the canonical model, but ran an additional set of experiments with beam search. Note that production systems will often perform beam search to find a more probable translation than one generated by greedy search.

Data: We trained all models with the German to English WMT De→En 2016 dataset (4,500,966 examples) (Bojar et al., 2016), validated with the WMT De→En 2015 development set (2,169 examples). We then used the WMT De→En 2016 test set (2,999 examples) to compute the hallucination percentage for each model.

For the input and output of all NMT models in consideration, we used sub-word tokens extracted by Byte-Pair Encoding (BPE) (Sennrich et al., 2015). To construct a vocabulary of unique tokens, we first combined the tokenized source and target corpora\footnote{We used Moses tokenizer: https://github.com/moses-smt/mosesdecoder/blob/master/scripts/tokenizer/tokenizer.perl}, and then learned a joint BPE code with an
Algorithm 1: Computing the percentage of hallucinations in a NMT model

Select a model; Fix a random seed;
Select a group of subword tokens with the following attributes:;
- Common tokens: 100 most common subword tokens;
- Mid-frequency tokens: random sample of 100 subword tokens between common and rare tokens;
- Rare tokens: 100 least common subword tokens;
- Punctuation: all punctuation tokens;

for every sentence in test corpus (e.g. WMT De→En 2016 test set) do
  if adjusted BLEU between reference sentence and translated sentence > 0.09 then
    for every selected token do
      for every location in (beginning, end, second-to-end, randomly in the middle) do
        put the selected token at the selected location in the byte-pair encoded input sequence;
        translate the perturbed input sequence;
        if adjusted BLEU between the translated, perturbed sentence and the translated, unperturbed sentence < 0.01 then
          this sentence can be perturbed to produce a hallucination;

8k merge operations budget, resulting in 12,564 unique tokens. Further, in order to study the effect of larger vocabulary sizes, we repeated the same process with 16,000 and 32,000 BPE codes and ended up with vocabularies having 19,708 and 36,548 unique tokens respectively and trained select models with larger vocabularies. Note that we used the same vocabulary for both source and target side languages.

4 Hallucinations

Informally, a hallucination is a translation of a perturbed input sentence that has almost no words in common with the translation of the unperturbed sentence. Here, perturb means adding a single token to the input source sequence. This is based on an initial observation that adding a rare token to an input sequence reliably caused a model to generate a hallucination, e.g. adding a Chinese character token to a German to English translation. We expanded and systematized this discovery into a brute force search procedure (Algorithm 1) by splitting our tokens into several types: common (100 most common German tokens), rare (100 least common German tokens), mid-frequency tokens (randomly sampled 100 tokens from the remaining German tokens), and punctuation tokens and perturbing each sentence by inserting a token at one of several positions: beginning, end, second to the end, and randomly in the middle. We did this for every sentence in our test set and collected statistics for each model variant.

To define a quantitative threshold for a hallucination, we adjusted the BLEU score, which is used to compare a reference sequence with a translated sequence. Briefly, the BLEU score is a common metric for translation, which measures weighted n-gram overlaps while penalizing short translations. We modified the BLEU score by re-weighting the n-grams in the BLEU score computation to favor having any words in common between the two sentences (1.0 for one-grams and 0.8 for bi-grams and disregarded other n-grams). Then, we call only sentences that have an adjusted BLEU score of less than 0.01 hallucinations. For examples of sentence pairs with different adjusted BLEU scores, see Section 8.2 in the Appendix.

Not all translations are good even before adding a perturbation token to the input sequence. We first excluded these poor translations by computing the adjusted BLEU score between the reference translation and the translation produced by the unperturbed input sequence. We kept sentences that had an adjusted BLEU of ≥ 0.09, a threshold that seemed to maintain enough context between the translation and the reference.
5 Hallucination Frequency and Reduction

5.1 Examples
We describe four common hallucination patterns: (1) grammatically correct output that bears no relation to the input text, (2) ungrammatical output with oscillatory structure, (3) output that remains largely in the source language, and finally (4) terse jumps to the end of the sequence. We also observe translations that are ungrammatical nonsense. While still highly undesirable, we note that a user should at least be able to spot and reject these additional hallucination patterns. See the Appendix section 8.3 for more examples.

Source: Freundschaft schließen durch Backen.
Reference: Make friends through baking.
Perturbation: Added ich: I randomly in sentence.
Grammatically correct hallucination: Should you want to join us?

Source: Monatelang war hinter verschlossenen Tren verhandelt, gegrübelt,
debattiert und gezeichnet worden.
Reference: Plans have been negotiated, mulled over, debated, and plotted behind closed doors for months.
Perturbation: Added uns: we at the beginning of the sentence.
Oscillatory hallucination: In the month of the month of the month of the month of the month, it was a matter of course.

Source: Neue Verhandlungen mit den Piloten
Reference: New negotiations with pilots
Perturbation: Added mit: with randomly in sentence.
Source language hallucination: Neu ist es mehr Jahren mit der Piloten d

Source: Fr die Fed-Repräsentanten beeinflussen die Marktturbulenzen die komplexe Kalkulation, wann man die Zinsen erhöhen solle.
Reference: For Fed policymakers, the market turmoil adds to the complex calculus of when to raise the interest rate.
Perturbation: Add uns: we randomly in sentence.
End of sequence hallucination: For FF.C.

5.2 Frequency of Hallucinations
We show that hallucinations can be easily evoked by inserting tokens in the source sequence. We used Algorithm 1 to quantify how susceptible a given model is to hallucination. In particular, we studied what types of perturbations (location, and token type) are more effective at inducing hallucinations. With this method, we found that, on average, 73% of all sentences in the WMT De→En test set can be perturbed to hallucination in the canonical model.

We studied how beam search, number of hidden units, vocabulary size, and decoding scheme affected hallucination percentages (Figure 1, left). We found that changing the number of hidden units to both 512 and 1024 from 256 and changing the vocabulary size—from 8K to 16K BPE codes did not significantly decrease the hallucination percentage. However, beam search and a vocabulary size increase corresponding to 32K BPE codes did significantly lower the mean percentage of hallucinations. We also studied how different types of perturbations impacted the hallucination percentage of the canonical model (Figure 1, right). By far, adding a perturbing token to the beginning of the input sequence induces the most hallucinations in the canonical model.

5.3 Reducing hallucination frequency
What changes can we make to the model to make it more robust to hallucinations? We investigated the effect of three different methodologies, (1) simple regularizations, (2) data augmentation and (3) regularizations on the dynamics in state space. We tested if a model variation significantly reduces hallucinations by performing a one-sided MannWhitney U between the canonical distribution of models and the distribution of models that use the model variant. We use a p-value of 0.05.

Simple Regularizations: We choose dropout, L2 regularization on embeddings (L2E) and L2 regularization on all weights (L2) as straight-forward regularization techniques to be applied. For dropout, we created a model with dropout in all feed-forward layers, with a keep probability of 0.9. Next, we implemented L2 regularization on the token-embeddings and throughout the entire model with weighting hyperparameters of $1 \times 10^{-4}$ and $1 \times 10^{-5}$, respectively.
Data augmentation (DA): We augmented the training data by perturbing all training sentences with a random token (either common, rare, mid-frequency, or punctuation) at either the beginning, end, second-to-end, or randomly in the middle while keeping the reference translation the same. This doubled our training set. We then trained a canonical model with the augmented training set, and found that data augmentation helps decrease hallucination percentages. We call this model DA.

Dynamical Regularizations: We wondered if hallucinations could be reduced by providing an initial state to the decoder that included context. So, we trained additional models where the initial state of the decoder was the last step of the encoder (TDIS). Note that the canonical model sets the decoder initial state as a vector of zeros. As a second regularization method that operates on the state space, we used Chaos-free network (CFN) (Laurent & von Brecht, 2016) which cannot produce chaos. We replaced the LSTM cell with the CFN in a set of experiments, again using 256 hidden units.

Dropout, L2E, and DA all cause statistically significant decreases in hallucination percentage, with DA being by far the most effective. On the contrary, switching out LSTM cells for CFN cells resulted in a significant increase, and TDIS had no statistical effect on the hallucination percentage.

Although data augmentation dramatically reduced hallucinations in the canonical model, it requires knowing the kind of perturbations used to induce a hallucination. To study how fine grained one’s knowledge must be, we trained the canonical model on a different training set where we withheld two types of data augmentation: perturbing at the beginning or with common tokens (We call this model DA w/o beginning or common). We then compared this model with the canonical model trained with the full DA training set (Figure 3). We found that DA w/o beginning or common perturbations yields much higher hallucination percentages when tested by perturbing at the beginning or with common tokens in comparison to the DA model. However, we also saw a reduction in hallucination percentage for common and beginning tokens when compared to the canonical model. This indicates that DA can still reduce hallucinations, even if exact perturbations are not known.
Figure 3: Effects of data augmented (DA) training when including all perturbation types vs excluding common and beginning perturbation types. We trained two models, one including all perturbations types for DA training, and the other excluding common and beginning perturbation types. We then examined the hallucination percentage of each perturbation type for both of these models and studied whether a DA model would be less prone to hallucinate when perturbed with types of tokens or positions it had not been trained against. Red star shows that DA w/o beginning or common had statistically significantly reduced mean compared to the canonical model trained without DA.

6 Analysis of hallucinations

6.1 Attention matrices

We observed a large difference between attention matrices of normal translations and of hallucinations. Attention networks in normal translations tend to study the entire input sequence throughout decoding. In French to English and other language pairs that are grammatically aligned (German to English is somewhat aligned), this often results a strong diagonal in the attention matrix. The attention matrix when translating hallucinations, however, shows the model attending to a few tokens. We give an example comparison in Figure 4, top panels. For additional attention matrices, see Section 9.

We wanted to quantify this difference in a way that does not require strong alignment between languages, i.e. one expects English to French to result in a largely diagonal matrix but not English to Turkish, so we used information entropy to compute a statistic that described the difference between attention matrices during a decode that resulted in a hallucination and those that resulted in a normal translation. Specifically, at each output of the decoder, the attention network gives a distribution over input tokens. We averaged these distributions across all decoded output tokens, resulting in a distribution of average attention weight over the input tokens. We treated this as a discrete distribution and computed the entropy, \( -\sum p(x_t) \log p(x_t) \), where \( x_t \) is the input token at time \( t \), for each example, resulting in a distribution of entropy values over all decoded sequences.

We then compared the entropy of average attention distributions between hallucinations and correct translations (Figure 4, bottom panels). This figure shows a significant difference between the entropy values for hallucination sequences. As a control, we show there is no significant difference between original input sequences and perturbed input sequences for sentences that cannot be perturbed to hallucination. Note that, in real world scenarios where a ground truth comparison between a reference translation and model translation is not possible, the measure of entropy of the average attention distribution may be useful to detect hallucinations.

The breakdown of the attention module seems to signal that the encoder and the decoder have been decoupled, and the decoder ignores context from the encoder and samples from its language model. Two possibilities are that broken attention modules are the root cause of decoupling, or they are a symptom of further breakdown in the dynamics of the decoder.
6.2 Stability analysis of the decoder

Examination of the types of hallucinatory text, oscillations and instability, led us to believe that hallucinations result from a dynamical process gone wrong in the decoder (whose cause may yet be further upstream). Further, many of the remedies to hallucinations can also be viewed as being dynamically regularizing. To study this idea more concretely, we defined a spectrum of stability exponents for the decoder and compared them between normal translations and hallucinations (Figure 5).

Concretely, we studied the stability of the decoder as a function of a given input token sequence, $x_{1:S}$ of length $S$. The sequence $x_{1:S}$ is run through the encoder, whose output is processed by the attention network, finally delivering an input to the decoder. For a given input token sequence, the decoder runs until it produces an end-of-sequence token, resulting in an output token sequence $y_{1:T}$ of length $T$ (or reaches a maximal decoded sequence length $T > 3S$). We were interested in studying $\frac{\partial h_T}{\partial h_0} = \frac{\partial h_T}{\partial h_{T-1}} \cdots \frac{\partial h_1}{\partial h_0}$ as many stability properties can be deduced from it. We note that if one is interested in studying $\frac{\partial h_T}{\partial x_s}$, the iterative process described by $\frac{\partial h_T}{\partial h_0}$ would still be critical to understand due to the chain rule.

We defined our spectrum of stability exponents, in analogy with Lyapunov exponents, but adapted for finite time by studying a finite-time version of Oseledets matrix, typically used in the study of chaotic dynamical systems. In particular, the $i^{th}$ stability exponent is defined as $\lambda_i(x_{1:S}) = \ldots$
\[
\frac{1}{2T} \log \left( \alpha_i(x_{1:S}) \right), \text{ where } \alpha_i(x_{1:S}) \text{ is the } i^{th} \text{ eigenvalue of the positive-semidefinite symmetric matrix } \\
\left( \frac{\partial h^T}{\partial h_0} (x_{1:S}) \right)^T \left( \frac{\partial h^T}{\partial h_0} (x_{1:S}) \right) \text{ and } h(t) \text{ is the decoder state at time } t \text{ concatenated across all layers. We used auto-differentiation software to exactly compute the Jacobian } \frac{\partial h^T}{\partial h_0} (x_{1:S}) \text{ so the complexities of the decoder circuitry were handled naturally (shown in Appendix, Section 8.1).}
\]

We show the distribution of stability exponents, comparing between all input sequences that could be made to hallucinate, and all those that could not (Figure 5). We show these for both the canonical model and the model trained with data augmentation. There are two observations. First, means of the distribution of the stability exponents for the canonical model, averaged over those sentences that could be perturbed to hallucinate, are statistically different than exponents averaged over sentences that could not be perturbed to hallucinate. Second, the distributions of the model trained with data augmentation show significantly reduced exponents in comparison to the canonical model.

![Figure 5: Stability analysis of hallucinations vs. normal translations.](image)

Methodologically, we note that it is typical to compute Lyapunov exponents using the algorithm of (Benettin et al., 1980). This is because studying the long-time behavior of dynamical systems typically requires many thousands of system iterations. We studied the numerical stability of our direct approach to computing our stability exponents using random chaotic RNNs (Sompolinsky et al., 1988) and found that the direct approach was more than adequate for short sequences and reasonable condition numbers, so we stayed with the simpler, direct approach.
7 Discussion

In this paper we uncovered and studied a hallucination-like phenomenon whereby adding a single additional token into the input sequence causes complete mistranslation. We showed that hallucinations are common in the NMT architecture we examined, as well as in its variants. We note that hallucinations appear to be model specific. We showed that the attention matrices associated with hallucinations were statistically different on average than those associated with input sentences that could not be perturbed. Finally we proposed a few methods to reduce the occurrence of hallucinations.

Our model has two differences from production systems. For practical reasons we studied a small model and used a limited amount of training data. Given these differences it is likely that our model shows more hallucinations than a quality production model. However, given news reports of strange translations in popular public translation systems (Christian, 2018), the dynamical nature of the phenomenon, the fact that input datasets are noisy and finite, and that our most effective technique for preventing hallucinations is a data augmentation technique that requires knowledge of hallucinations, it would be surprising to discover that hallucinations did not occur in production systems.

While it is not entirely clear what should happen when a perturbing input token is added to an input source sequence, it seems clear that having an utterly incorrect translation is not desirable. This phenomenon appeared to us like a dynamical problem. Here are two speculative hypotheses: perhaps a small problem in the decoder is amplified via iteration into a much larger problem. Alternatively, perhaps the perturbing token places the decoder state in a poorly trained part of state space, the dynamics jump around wildly for while until an essentially random well-trodden stable trajectory is found, producing the remaining intelligible sentence fragment.

Many of our results can be interpreted from the vantage of dynamical systems as well. For example, we note that the NMT networks using CFN recurrent modules were highly susceptible to perturbations in our experiments. This result highlights the difficulty of understanding or fixing problems in recurrent networks. Because the CFN is embedded in a larger graph that contains an auto-regressive loop, there is no guarantee that the chaos-free property of the CFN will transfer to the larger graph. The techniques we used to reduce hallucinations can also be interpreted as dynamical regularization. For example, L2 weight decay is often discussed in the context of generalization. However, for RNNs L2 regularization can also be thought of as dynamically conditioning a network to be more stable. L2 regularization of input embeddings likely means that rare tokens will have optimization pressure to reduce the norm of those embeddings. Thus, when rare tokens are inserted into an input token sequence, the effects may be reduced. Even the data augmentation technique appears to have stability effects, as Figure 5 shows the overall stability exponents are reduced when data augmentation is used.

Given our experimental results, do we have any recommendations for those that engineer and maintain production NMT systems? Our results on reducing hallucinations suggest that standard regularization techniques such as Dropout and L2 weight decay on the embeddings are important. Further, data augmentation seems critical and we recommend inserting randomly chosen perturbative tokens in the input sentence as a part of the standard training regime (while monitoring that the BLEU score does not fall). We note a downside of data augmentation is that, to some extent, it requires knowing the types of the pathological phenomenon one desires to train against.

References


8 appendix

8.1 NMT Decoder Schematic

Figure 6: Schematic of the NMT decoder. The input sequence, $\mathbf{x}_{1:S}$, is encoded by a bidirectional encoder (not shown) into a sequence of encodings, $\mathbf{z}_{1:S}$. The attention network, $f_{\text{att}}$, computes a weighted sum of these encodings (computed weights not shown), based on conditioning information from $\mathbf{h}$ and provides the weighted encoding to the 2-layer decoder, $f_{\text{dec}}$, as indicated by the arrows. The decoder proceeds forward in time producing the translation one step at a time. As the decoder proceeds forward, it interacts with both the attention network and also receives as input the decoded output symbol from the previous time step.

8.2 Adjusted BLEU score

Examples of pairs of sentences with different adjusted BLEU scores are as follows:

- **BLEU: 0.5**
  Sent 1: The role you play when creating the news is very important.
  Sent 2: The part you play in making the news is very important.

- **BLEU: 0.09**
  Sent 1: At the moment, men are overweighed by men.
  Sent 2: Currently the majority of staff are men.

- **BLEU: 0.05**
  Sent 1: Austria has also introduced controls to its southern and eastern boundaries.
  Sent 2: The German government has also established the most recent and national and international government institutions.

- **BLEU: 0.01**
  Sent 1: “But then, suddenly, our small Jumbo was there,” says Brigitte Probst.
  Sent 2: Scientists, who are not in the right place, are not in the right place.

- **BLEU: 0.001**
  Sent 1: Mr Corbyn said that Angela Eagle, the shadow of Economic Minister, will be represented in the economic minister, when he does not participate—for example, when Mr. Cameron is on the
  Sent 2: Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr. Mr.
As seen above, an adjusted BLEU score of $< 0.01$ means the two sentences have very few words in common.

### 8.3 Additional Examples

Additional examples of perturbed input sentences. For many sentences, there were more than one combination of token and position that caused a hallucination. In those cases, we give several examples of different perturbations and their hallucinations.

'@@' indicates a subword token. For example, perturbing the beginning of the sentence "Vor elf..." with g@@ would give: "gVor elf..."
Beginnend: Ich: Don’t hesitate to contact us at any time of your visit.
568  Wir: d Laut den Medienberfre sei jederzeit einen ersten Bus einem ersten Bus mit rund 50 Rechtlingen im
569  Serbischen Grenzstadt Sid, im Lauf der Tage would der weiteren Busse a/e/s>
570  Beginnend: Kann: Don’t hesitate to contact us at any time.

571  Source: Neue Verhandlungen mit den Piloten
572  Reference: New negotiations with pilots
573  NMT Translation: New negotiations with the pilots
574  Beginnend: die: dass Neue News bis die Piloten
575  Beginnend: auf: la Neue Neuere negations mit den Piloten d
576  Beginnend: einen: en Neue Negotions mit den Piloten d
577  Beginnend: die: de Neue Neuere negations mit den Piloten
den:
578  Beginnend: kann: de Neue Neuere negations mit den Piloten
die:
579  Beginnend: durch: durch Neue Neuere negations mit den Piloten
die:
580  Beginnend: seit: Seit e mehr Jahren mit der Piloten d
581  Randomly: kann: kann die Neue Neuere negations mit den Piloten
die:
582  Beginnend: ihre: die Neue Neuere negations mit den Piloten
die:
583  Randomly: um: um die Neue Neuere negations mit den Piloten
die:
584  Randomly: in: in die Neue Neuere negations mit den Piloten
die:
585  Randomly: durch: durch die Neue Neuere negations mit den Piloten
die:

586  Source: Fr die Fed-Representanten beeinflussen die Marktsturbulenzen die komplexe Kalkulation, wann man die Zinsen
587  erhöhen soll.
588  Reference: For Fed policymakers, the market turmoil adds to the complex calculus of when to raise the interest rate.
589  NMT Translation: For the Fed-Representatives, the market turbos influence the complex calculation, when we increase the
590  interest rate.
591  Randomly: uns: For FF.C.
592  Source: Schon lnger war bekannt, dass es unter der Eisdecke von Enceladus fliessiges Wasser geben muss.
593  Reference: It has long been known that there must be liquid water under Enceladus "apos; s crust of ice.
594  NMT Translation: It was already well known that there must be fluid water under the ice cup of Enceladus.
595  Beginnend: Ich: I am looking forward to hearing from you.
596  Source: Freundschaft schlieen durch Backen.
597  Reference: Make friends through baking.
598  NMT Translation: Friendship through baking.
599  Randomly: ich: Should you want to join us?
600  Source: In der Gesellschaft hat sich – auch wegen des Fachkrftemangels – mehrheitlich die Sichtweise durchgesetzt, dass
601  Flchtlingen schneller der Zugang zum Arbeitsmarkt ermglicht werden sollte.
602  Reference: As far as the majority of society is now concerned, particularly in view of skills shortage, refugees should be
603  allowed to gain faster access to the job market.
604  NMT Translation: Most of the time in the company – even because of the lack of expert labour – a majority of the view that
605  refugees should be made faster from access to the labour market.
606  Beginnend: Unfortunatly, it is not possible to have a look at the situation.
607  Source: Monatelang war hinter verschlossenen Tren verhandelt, gegrbelt, debattiert und gezeichnet worden.
608  Reference: Plans have been negotiated, mulled over, debated, and plotted behind closed doors.
609  NMT Translation: Monatelang has been negotiating, debated and drawn behind closed doors.
610  Beginnend: Monat lang: In the month of the month of the month of the month of the month, it was a matter of course.
611  Beginnend: uns: During the month of the month of the month of the month, it was a matter of course that it is not possible for us
612  to take a look.
613  Randomly: Moat: During the month of the month of the month of the month, it was not possible to have the opportunity to take a look
614  at the situation.
615  Randomly: Noat: During the month of the month of the month of the month, it was not possible to have a look at the situation.

9 Additional Attention Matrices

9.1
Figure 7: Example attention matrices. (left) Attention matrix for the original input sentence. (right) Attention matrix for the perturbed input sentence. All translations of the perturbed input sentence shown here are hallucinations. All decoding was done with the canonical model.