A Study on Hallucination and its Detection Methods

Nesara Eranna Bethur
Georgia Institute of Technology
nbethur3@gatech.edu

Angana Borah
Georgia Institute of Technology
aborah7@gatech.edu

Pratyusha Maiti
Georgia Institute of Technology
pmaiti6@gatech.edu

Abstract

In this work, we study hallucinations in neural machine translation (NMT), which lie at the extreme end of machine translation pathologies. Initially, we use perturbation methods to induce hallucination in translation tasks and understand the magnitude of performance changes in NMT models, using evaluation metrics like BLEU and METEOR. We next device hallucination detection via text classification and NMT. First, for the text classification-based hallucination detection task, we use BERT and analyze its performance using explainability techniques like Integrated Gradients. This motivates us to approach data manipulation techniques like tagging, perturbation, and prompt engineering mechanisms. Finally, for the NMT-based hallucination detection, we use an unsupervised approach to detect hallucinations by utilizing attention matrices in translation models. Upon experimentation, we find that perturbation using misspelled words leads to high degradation in translation performance for NMT models (46% reduction in BLEU score). Additionally, data manipulation, prompt engineering, and the use of attention lead to better performances in neural models.

1. Introduction

In the age of generative Artificial Intelligence (AI), there has been a huge demand in creation and training of Large Language Models (LLMs). With frameworks like ChatGPT, Bard etc. the general public has access to LLMs which may generate hallucinatory content.

A generated content is said to be hallucinatory if the generated output either contradicts the source input or cannot be verified from the source content [7]. The phenomenon of hallucinations is observed often with LLMs.

Hallucinations may result in derogatory content and discrimination which may cause harm to society. So, by solving the problem of hallucination, our society benefits. The LLMs without hallucinations will be more faithful, fair, and devoid of unnecessary biases.

With the motivation to understand and detect hallucinations, we compile a study on hallucination and its detection methods. We specifically address the problem of hallucinations in Neural Machine Translation (NMT).

The key contributions of our work are as follows:

- In our work, we understand hallucinations through a quantitative analysis with different metrics, like BLEU and METEOR, which have higher correlations with human judgments.
- We develop a text-classification-based supervised hallucination detection and visualization framework to observe why models hallucinate. We report interesting observations which strengthen the importance of data. So, we exploit prompt engineering, data manipulation and perturbation to detect hallucinations better. We report the improvements in accuracy to strengthen our observations.
- We explore the correlation between the sparsity of source attention matrices and the distance between the source and target attention matrices with the occurrence of hallucinations in translation tasks. Based on the findings, we propose an Neural Machine Translation-based unsupervised hallucination detection model that uses optimal transport distance between attention matrices as the loss function.

2. Background

Current research is focused on the hallucination mitigation by having a threshold for the BLEU scores [8]. Hallucination induction is performed by perturbing the data with a few high-frequency words at different positions in the sentence. With this setup, various data augmentation techniques are performed to analyze their effects on the resulting hallucinations.

There also has been work trying to correlate memorization and hallucination [11]. Memorization Value Estimator from [3] is used to observe that memorized words result in more hallucinatory content when compared to normal works.

All these works assume BLEU scores to be the golden standard to evaluate hallucinations. A robust visualization framework is also absent in these works. We take these limitations as our starting point for our work.

To mitigate hallucinations, few works propose model architecture changes. [6] and [2] propose a dual encoder structure that provides additional context for the model. [14] explores the option of having a sparsely connected attention layer with an inductive bias to improve the model performance with respect to hallucinations. [13] proposes a dual decoder structure to reduce hallucinations.

From our visualization framework, we observe the need for data-centric approaches to detect hallucinations. So we lean more towards data-centric over model-centric approaches to improve hallucination detection.

Recent studies by Guerreiro et al [4] show that statistical differences in the encoder-decoder attention patterns are indicative of hallucinations in high-quality translations. This can be formulated as an optimal transport problem and can be implemented as an unsupervised, plug-in detector that can be used with any attention-based translation model. These findings indicate that the sparsity of attention matrices and the distance between source and target attention matrices can be used as a loss function to train a hallucination detection model.

3. Dataset information

For our experiments, we use the dataset by [5], which contains German to English translations with respective labels for hallucinations, namely, hallucinations related to repetitions, named-entities, omission, strong-unsupport, and full-unsupport. We add the label 'hallucination', which denotes the presence of hallucination if either of the above hallucination types is present[1]. The dataset has 3415 examples in total. Table 1 represents the dataset statistics.

For our unsupervised hallucination detection experiments, we use an annotated dataset of 423 translation examples[2].

Label	**Total examples**
Non-hallucinated	2883
Repetitions	87
Named-entities	41
Omission	204
Strong-unsupport	164
Full-unsupport	129
Hallucinated	532

Table 1. Dataset statistics

4. Approach

4.1. Understanding Hallucinations

We use a $MarianMT$ model, from $simpletransformers$[3] library to perform neural machine translation. We specify the $encoder\_decoder\_type$ as "marian", and use the architectures and trained weights from Huggingface's $Helsinki-NLP/opus-mt-de-en$ model.

To understand the magnitude of performance degradation in BERT due to hallucinations, we perturb the source data to induce hallucinations in the dataset. For the dataset, we choose only the non-hallucinated examples, thus containing 2883 examples. We then use a split of 80/10/10 into train/dev/test.

We randomly choose 50% of each of the train, validation, and test data and add perturbations: random token insertion at the beginning (using most frequent and least frequent tokens), and random misspelling of words (by random deleting a character in a random word in a string) in our source data. We finetune the MarianMT model on this dataset for 10 epochs and report the results on the test data.

4.2. Hallucination Detection via Text Classification

Hallucination detection is the task of detecting hallucinations in a given text. We use $bert - base - multilingual - uncased$ from $Huggingface$, with an AdamW optimizer, using a learning rate of $4e - 5$, an epsilon of $1e - 8$, and a weight decay of 0.003. We finetune our model for 4 epochs. We do a train/dev/test split of 80:10:10 on our dataset. The input to our model is the concatenated source and target strings, and the output is the hallucination label.

4.2.1 Handling data imbalance

Since the dataset contains 2883 non-hallucinated examples and only 532 hallucinated examples, we also perform experiments by oversampling and undersampling our data. Undersampling involves the removal of non-hallucinated examples randomly so that we have an equal number of hallucinated and non-hallucinated examples. Whereas oversam-

[1] We add the 'hallucination' label for an easier hallucination detection task, since the number of different types of hallucinations in the dataset in limited

[2] https://github.com/weijia-xu/hallucinations-in-nmt/blob/main/data/eval.deen.tsv

[3] https://simpletransformers.ai/docs/seq2seq-model/

pling involves randomly duplicating hallucinated examples so that they are equal to non-hallucinated examples.

4.2.2 Integrated Gradients for visualization

To understand the performance of our BERT model better, we utilize integrated gradients. It is a powerful axiomatic attribution method that requires almost no modification of the original network. It helps us look a the most important tokens that BERT utilizes in determining the output class, by computing the attribution for all the individual neurons in the embedding layer and calculating the salience score for each token by averaging the attributions over the embedding dimension. We use the $Captum$[4] library for integrated gradients for BERT finetuned on oversampled and undersampled test data.

4.2.3 Data-centric methods to improve Hallucination Detection

Perturbation:
We add certain perturbations to our target data to include more hallucinated examples, for eg., adding random tokens (most frequent and least frequent tokens from the dataset) at the beginning of the sentences, and mis-spelling random words by deleting some characters. This is inspired by [12], who found perturbations are important for understanding hallucinations.

Tagging:
We also explore the addition of finer tags to the data to improve BERT performance. We apply three different tagging approaches:

1. ”SRC” and ”TGT” tags: Instead of simply concatenating source and target strings, we add tags ”SRC: ” and ”TGT: ” in front of source and target strings respectively before concatenating.

2. PoS tags: We add part of speech tags before every word in a string, for eg. The sentence ”Noah lives in Berlin”, would be changed to ”PROPN_Noah VERB_lives ADP_in PROPN_Berlin”.

3. NE tags: We add named entity tags to named entities in the string, for eg. The sentence ”Noah lives in Berlin”, would be changed to ”PER_Noah lives in LOC_Berlin”.

We use the $SpaCy$[5] library for PoS and NER tagging for both our German source data and English target data.

Prompt Engineering:
Taking inspiration from several prompt engineering frameworks [9], we try to build a train prompt like the following for our detection model:

”German: Also, es ist nicht nur ein Verzeichnis. English: So it is not just a directory. Hallucination : 0”

The above goes in as an input string to the BERT model, with the hallucination as the label (like earlier). However, for the validation and test sets, we do not include the hallucination label. The validation and test set input feature string has the following structure:

”German: Also, es ist nicht nur ein Verzeichnis. English: So it is not just a directory. Hallucination : ?”

4.3. Hallucination Detection via Neural Machine Translation

The problem of hallucination detection can also be addressed by the intuition that since hallucinations contain content strongly detached from the source, they may exhibit encoder-decoder patterns that are statistically different from high-quality translations. Specifically, if the attention matrices of source texts are sparse, it denotes that fewer source tokens are used for translation and may lead to hallucinated content. For this, we use a simple Seq2seq model with two pre-trained BERT models for the WMT14English-German machine translation task where we take $bert-base-cased$ as the encoder and $bert-base-german-cased$ as the decoder. We use an annotated dataset of 423 translation examples. The attention matrices are observed for sparsity in case of hallucinations.

Based on the findings, we use a plug-in detector based on the $Wasserstein$ distance to estimate the cost of transforming a source distribution (attention matrices) into a target distribution. Higher the cost of the translation, the more distant will be the translated content from the source. We use this detector logic as a loss function, with an AdamW optimizer, ReduceLROnPlateau scheduler, learning rate of 1e-5. We finetune the model for 5 epochs with a train-test ration of 80:20. The aim is to train an adapter module for any attention-based model to minimize occurrence of hallucinations in machine translation tasks.

5. Experiments and Results

5.1. Understanding hallucinations

We employ two metrics to evaluate our translation results: BLEU and METEOR. BLEU [10] is a widely employed metric for machine translation evaluation. METEOR [1] is a newer metric, which takes uses recall in addition to precision while BLEU only uses precision.

[4] https://captum.ai/docs/extension/integrated_gradients
[5] https://spacy.io/

Data	**BLEU score**	**METEOR score**
Original data	8.87	0.40
50% misspelled data	4.77	0.05
50% random word insertions on data	8.6	0.39

Table 2. Hallucination Induction results

Data	**Test accuracy**
Original	85.8%
Oversampling	73.1%
Undersampling	53.5%

Table 3. Hallucination Detection results with different sample sizes

METEOR shows a better correlation with human judgment than BLEU. Table 2 reports results for translation by the MarianMT model finetuned on our data. We employ *corpus_bleu* and *meteor_score*, both available in the *nltk*[6] library. We see that, although the performances of the fine-tuned model are not very high, there is a significant drop in performance for the model which is fine-tuned on the 50% misspelled dataset. A lower BLEU score might suggest stronger hallucinations in results [12]. It is also interesting to note that adding random words on 50% of the data does not change the performance of the translation. This is an interesting indication that this might not be a very important factor for inducing hallucinations.

5.2. Hallucination Detection via Text-classification

Table 3 contains the results for hallucination detection via text classification on oversampled and undersampled datasets. Although BERT has the highest accuracy for original data, it is because it classifies all data points as non-hallucinatory (due to the presence of data imbalance in our dataset). Out of the other three cases, BERT performs the best in terms of oversampling. This is understandable because it has the highest number of data points to learn from.

Token	**Avg Attribute Score**	**Token**	**Avg Attribute Score**
event	0.960	##chen	0.358
allowing	0.923	markt	0.340
site	0.918	environmental	0.337
bad	0.917	china	0.327
##ter	0.865	##fni	0.327
javascript	0.811	workers	0.320
slot	0.796	markets	0.298
##ful	0.750	6	0.292
terrain	0.720	##lung	0.282
room	0.691	charlie	0.273
gentlemen	0.674	##zusetzen	0.269
##able	0.668	##gang	0.266
stehen	0.643	##bs	0.259
commissioner	0.639	300	0.246
##inische	0.576	selection	0.239
council	0.560	fluid	0.235
see	0.555	neu	0.228
votes	0.545	final	0.216
mensch	0.540	maintenance	0.216
vat	0.531	mann	0.215

Table 4. Integrated Gradients results (left for undersampled data and right for oversampled data)

Table 4 contains the integrated gradients for BERT fine-tuned on oversampled and undersampled datasets. We do not find any deducible reasons for the tokens for the undersampled dataset. However, for the oversampled dataset, we find named entities like 'China', 'Charlie', and numerics like '6', and '300', which are important entities for hallucination ([7, 5]). There are also several German words and subwords as top tokens. For example, *##chen* may be a subword for *Schengen*, *Europäichen* (from words that are present in the dataset), which are both named entities. *##gang* subword in German is mostly used in path or way (passage, corridor, etc.), and maybe a subword for words like durchgang, eingang, etc., which may in turn refer to locations. This suggests that a model that pays higher importance to named entities, numerics, etc. may perform well on hallucination detection.

Finally, Table 5 shows the results of the hallucination detection using different data perturbation, tagging and prompt engineering approaches. We apply these approaches to our undersampled dataset. We see that BERT has difficulties learning in the case of perturbed dataset, which may be because it introduces further variation in our dataset. However, we believe that given a larger dataset, adding data perturbation can help BERT perform better. We find that just adding "SRC" and "TGT" tags improves the performance a lot. Moreover, both addition of PoS and NE tags also slightly improves the performance on the test set. Finally, prompt engineering helps BERT in achieving better accuracy. This shows that, adding these features to the data and better prompting can assist BERT in the hallucination detection task. We believe introducing a larger dataset for fine-tuning might further improve the results.

5.3. Hallucination detection via Neural Machine Translation

Table 6 shows that sparsity of attention matrix for the source is slightly but inconclusively higher in case of hallucinated content. The sparsity is calculated as the ratio of the number of non-zero items in the diagonal of the attention matrix by the total number of items. This shows that there

[6] https://www.nltk.org/

Tagging	Test accuracy
Perturbed	56.4%
SRC and TGT tagging	63.8%
PoS tagging	55.8%
NE tagging	54.5%
Prompt engineering	56.8%

Table 5. Data-centric approaches to improve Text-classification-based hallucination detection.

Label	Sparsity
Non-hallucinated	0.0377
Hallucinated	0.0554

Table 6. Source Attention Sparsity in Hallucinated vs Non-Hallucinated Translations

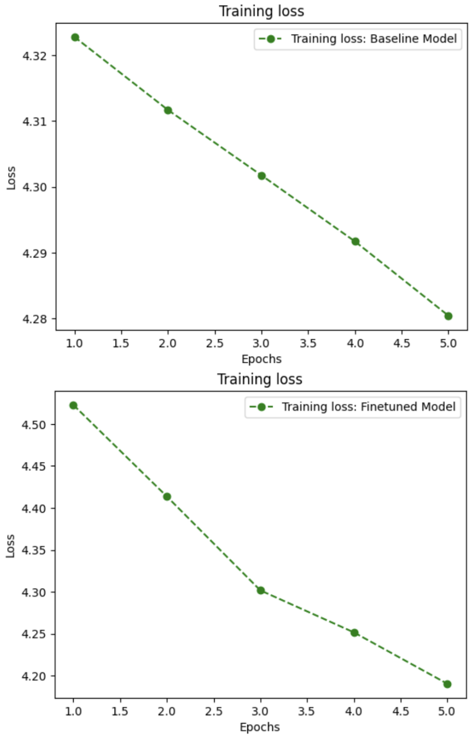

Figure 1. Training loss for baseline vs fine-tuned model.

Data	BLEU score	METEOR score
Baseline	5.432	0.20
Finetuned	5.977	0.22

Table 7. W-dist adapter fine-tuned vs baseline BLEU and METEOR scores

may be some but not a strong correlation between hallucinations and how few of the source tokens are attended to during the encoding process.

Next, we train a Seq2seq model with the hallucination detector adapter plugged in and compare it's performance with respect to the baseline model which is the same architecture, without the adapter plugged in. Figure 1 shows the losses for the baseline and the fine-tuned models over 5 epochs. Since we have an additional w-dist loss with the adapter plugged in, we can see an initial higher loss when training with the adapter but we see a faster convergence suggesting that the model learns translations faster with the adapter plugged in.

Table 7 shows the BLEU and METEOR scores for each of the models. We see that the additional w-dist loss results in marginal improvement in performance on translation task in terms of both BLEU and METEOR scores. Since we have used a simple Seq2seq model with two pre-trained BERT models as encoder-decoders over a small dataset, we do not expect the scores to match the state of the art results but the results show potential in terms of using attention scores as a means of detecting and mitigating hallucinations in LLMs.

6. Limitations and Future Work

A major limitation of our project is that our datasets are imbalanced and small, and hence, we want to experiment with larger datasets in the future. However, a smaller dataset also gave us an opportunity to try different data manipulation and perturbation techniques, which might have been difficult in a larger dataset. In the future, we are interested to look at different types of hallucinations and analyze the performance of neural models on them. An important research direction is also to build an annotated dataset of hallucinations since current datasets (that we know of) are limited in size.

7. Conclusion

In our work, we showed that data manipulation and perturbation are important to improve the performance of neural models for hallucination detection. We also showed that models that pay attention to certain tokens like named entities, and numerics tend to perform better in the hallucination detection task. Finally, using sparsity and the distance between source and target attention matrices as a loss function resulted in a marginal improvement in the translation performance, which shows the importance of using attention scores for understanding hallucinations. Our code and dataset are available here: https://github.com/AnganaB/Hallucinations_NMT.

We hope that our work serves as a useful step towards understanding and mitigating hallucinations in neural models.

8. Work Division

Table 8 shows the contributions of team members.

Student Name	Contributed Aspects	Details
Nesara Bethur	Data Creation, Implementation and Analysis	Trained MarianMT models on different datasets (including perturbed sets), and evaluated translation performance using different metrics. Worked on prompt engineering frameworks for hallucination detection.
Angana Borah	Data Creation, Implementation and Analysis	Trained BERT model, and implemented data changes (including perturbation and prompt engineering) for hallucination detection. Computed Integrated Gradients and analyzed the performance of BERT on different datasets.
Pratyusha Maiti	Implementation and Analysis	Trained the adapter-infused seq2seq model and analyzed the results of attention-based hallucination detection methods.

Table 8. Contributions of team members.

References

[1] Satanjeev Banerjee and Alon Lavie. 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, June 2005. Association for Computational Linguistics. 3

[2] Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. Faithful to the original: Fact aware neural abstractive summarization, 2017. 2

[3] Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 2881–2891. Curran Associates, Inc., 2020. 2

[4] Nuno M. Guerreiro, Pierre Colombo, Pablo Piantanida, and André F. T. Martins. Optimal transport for unsupervised hallucination detection in neural machine translation, 2022. 2

[5] Nuno M Guerreiro, Elena Voita, and André FT Martins. Looking for a needle in a haystack: A comprehensive study of hallucinations in neural machine translation. *arXiv preprint arXiv:2208.05309*, 2022. 2, 4

[6] Luyang Huang, Lingfei Wu, and Lu Wang. Knowledge graph-augmented abstractive summarization with semantic-driven cloze reward. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5094–5107, Online, July 2020. Association for Computational Linguistics. 2

[7] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12), mar 2023. 1, 4

[8] Katherine Lee, Orhan Firat, Ashish Agarwal, Clara Fannjiang, and David Sussillo. Hallucinations in neural machine translation, 2019. 1

[9] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35, 2023. 3

[10] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 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, July 2002. Association for Computational Linguistics. 3

[11] Vikas Raunak, Arul Menezes, and Marcin Junczys-Dowmunt. The curious case of hallucinations in neural machine translation, 2021. 2

[12] Vikas Raunak, Arul Menezes, and Marcin Junczys-Dowmunt. The curious case of hallucinations in neural machine translation. *arXiv preprint arXiv:2104.06683*, 2021. 3, 4

[13] Kaiqiang Song, Logan Lebanoff, Qipeng Guo, Xipeng Qiu, Xiangyang Xue, Chen Li, Dong Yu, and Fei Liu. Joint parsing and generation for abstractive summarization, 2019. 2

[14] Zeqiu Wu, Michel Galley, Chris Brockett, Yizhe Zhang, Xiang Gao, Chris Quirk, Rik Koncel-Kedziorski, Jianfeng Gao, Hannaneh Hajishirzi, Mari Ostendorf, and Bill Dolan. A controllable model of grounded response generation, 2021. 2