How to decrease LLMs' hallucinations: a proposal

Tatiana Ilina, Martine Ceberio, Marcelo Frias, Christoph Lauter,
Vladik Kreinovich, and Nguyen Hoang Phuong

Abstract It is well known that LLMs' hallucinations are a big problem. Usually, we detect hallucinations by noticing that the LLM's answer is inconsistent with some well-established rules or facts. From this viewpoint, a natural way to decrease hallucinations is as follows: every time we have new pairs (input, output) on which to train, we should also again train on the pairs (input, output) corresponding to well-established facts and rules.

Tatiana Ilina
Selection Lab, Greenfield Park, Quebec, Canada, e-mail: tatiana@theselectionlab.ai

Martine Ceberio
Department of Computer Science, University of Texas at El Paso, 500 W. University
El Paso, Texas 79968, USA, e-mail: mceberio@utep.edu

Marcelo Frias
Department of Computer Science, University of Texas at El Paso, 500 W. University
El Paso, Texas 79968, USA, e-mail: mfrias4@utep.edu

Christoph Lauter
Department of Computer Science, University of Texas at El Paso, 500 W. University
El Paso, Texas 79968, USA, e-mail: cqlauter@utep.edu

Vladik Kreinovich
Department of Computer Science, University of Texas at El Paso, 500 W. University
El Paso, Texas 79968, USA, e-mail: vladik@utep.edu

Nguyen Hoang Phuong
Artificial Intelligence Division, Information Technology Faculty, Thang Long University
Nghiem Xuan Yem Road, Hoang Mai District, Hanoi, Vietnam
e-mail: nhphuong2008@gmail.com

1 Formulation of the problem

It is well known that Large Language Models (LLMs) – such as the most well-known ChatGPT – often produce completely wrong answers to the user queries. Such answers are called *hallucinations*; see, e.g., [1] and references therein. For example, when asked to find papers on a certain topic, LLM sometimes produced fake citations to non-existing journals and/or non-existing papers in known journals.

When ChatGPT was launched, 15 to 20 percent of the answers were hallucinations. At present, this percentage is down to 5%. This is much smaller, but still, for many applications, this is unacceptable. For example, in medical application, it would mean that one out of every 20 patients will het a wrong recommendation and maybe die.

We therefore need to drastically reduce the number of hallucinations. How can we do it?

2 Analysis of the problem

How we can usually detect hallucinations. To start solving the hallucination problem, let us recall how we know that a certain computer answer is a hallucination. Usually, the answer is recognized as a hallucination if it contradicts some well-established fact or rule. For example, we can easily check that an alleged paper in a known journal is indeed fake if a search in the journal's table of contents does not find this alleged paper.

In other cases, the detection is somewhat more complicated, but it still eventually boils down to a contradiction with a well-established fact or rule. Let us give two examples when one of us (VK) easily detected hallucinations in LLM-produced answers to a google query.

In the first of these examples, the query was about how many people with PhD are there in the world. The LLM-produced answer was 8 million, or 1% of the world's population. This cannot be right, since it is well established that about 8 billion people live in the world, and 1% of this amount is 80 million – much larger than 8 million.

Another example is when the query asked how faster is a (more expensive) express-train ride from Dusseldorf to Hannover than a (less expensive) ride by regular trains. The LLM's answer was that it is much faster: 2 hours 37 minutes by express trains versus 2 hours 50 minutes by regular trains. This answer is a hallucination since it contradicts a well-established fact that 2 hours 37 minutes is faster – but *not* much faster – than 2 hours 50 minutes.

But why do LLM's answers often contradict well-established knowledge – while human answers contradict it in a much fewer cases? In our opinion, the main reason for this difference is that humans have a set of well-established facts and rules that is very difficult to change. In contrast, in LLMs, all the weights are

subject to the same types of changes, there is no difference between accidental facts and well-established facts. When we learn new knowledge, all weights change – both weights leading to accidental facts and weights leading to well-established facts. As the LLM learns and weights change, what changes is not only the answers to queries related to the new knowledge, but also (the previously perfect) answers to well-established queries.

This analysis leads to the following proposal.

3 Our proposal

Our idea. Our main idea is to specifically select a list L of queries for which there is a well-established answer. These may be well-established facts about which we have no doubt – e.g., that there are 8 billion people ion the world, that 1% of this amouns is 80 million, etc. These may be particular cases of inference rules: e.g., statements like "if A implies B and B implies C, then A implies C" for different A, B, and C.

Then, after initial training, when we use new pairs (question, answer) to teach new things to LLM:

- the traditional approach is train the neural network (NN) behind the LLM with the new pairs, while
- the proposed approach is to train the NN both on the new pairs *and* on the pairs from the set L of well-established facts and rules.

In the proposed training, it is also important to make sure that the objective function and stopping criteria treat the pairs of these two types differently:

- for the new pairs – corresponding to the new knowledge – we should avoid overfitting, we should stop training when there is some difference between the desired numerical answer y_i and the answer $\widetilde{y}_i$ generated by the NN;
- in contrast, for the pairs (X_j, Y_j) corresponding to well-established knowledge, the NN's answer $\widetilde{Y}_j$ should be effectively equal to the desired one,

This difference can be implement by giving new pairs (x_i, y_i) much smaller weight than the pairs (X_j, Y_j) corresponding to the well-known facts and rules. For example, the least squares objective function should have the following form:

$$\sum_i \frac{(y_i - \widetilde{y}_i)^2}{\sigma^2} + \sum_j \frac{\left(Y_j - \widetilde{Y}_j\right)^2}{\delta^2},$$

where δ is much smaller than σ: $\delta \ll \sigma$.

Will this work – and answer to a possible objective. At first glance, it may seem that this approach will not work: when we train the NN on the old data, the weights

reduce to what we had before, and as a result, there will be no learning. However, a more detailed analysis shows that this objective can be easily answered.

Indeed, in the current successful deep neural networks, the number of weights is several magnitudes larger than the number of inputs. As a result, when we try to find the weights that correctly works on all new training pairs, we have much fewer equations than unknowns. As a result, this problem has many possible solutions, i.e., there are many different combinations of weights that provide close-to-perfect learning. The weights from different combinations can be drastically different – while each of these combinations leads to an almost perfect fit with the new pairs.

Since neural networks are universal approximators, among these almost-perfectly-tinning-new-pairs combinations there are some that also fit the pairs from the set L of what we called well-established facts and rules. The difference between the results of the traditional and the newly proposed training is that:

- as a result of the traditional training, we get a combination of weights that almost perfectly fits the new pairs, but that may lead to not so good fit for pairs from the set L;
- in contrast, as a result or the proposed training, we get a combination of weights that not only alsmost perfectly fits the new pairs, but also fits all the pairs from the set L – i.e., does not forget the well-established facts and rules.

Limitation of the proposed approach. The main limitation of the proposed approach it that it makes training longer, since we now have more pairs to train – in addition to the new pairs, we have to also used "old" pairs corresponding to well-established facta dn rules. This is probabily not as bad as it sounds, since training LLMs requires a lot of time anyway.

Acknowledgments

This work was supported:

- by the AT&T Fellowship in Information Technology,
- by the Institute for Risk and Reliability, Leibniz Universitaet Hannover, Germany,
- by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) Focus Program SPP 100+ 2388, Grant Nr. 501624329,
- by the European Union under the project ROBOPROX (No. CZ.02.01.01/00/22 008/0004590),
- by the Center of Excellence in Econometrics, Faculty of Economics, Chiang Mai University, Thailand,
- by the Ho Chi Minh City University of Banking, Vietnam, and
- by Thang Long University, Hanoi, Vietnam.

References

1. S. Greengard, "Shining a light on AI hallucinations: how they come about, their impact, and what can be done to mitigate or entirely end artificial intelligence's tendency to hallucinate", *Communications of the ACM*, 2025, Vol. 68, No. 5, pp. 9–11, DOI 10.1145/3715691