

# KnowHalu

KnowHalu introduces a pioneering approach to hallucination detection in AI-generated content, focusing on enhancing the reliability and factual accuracy of language models 🧠. Our work is structured around a comprehensive pipeline designed to identify and rectify hallucinations through a multi-stage factual checking process 🔧.

The hallucination detection process begins with **Non-Fabrication Hallucination Checking**, a preliminary phase aimed at identifying hallucinations based on the specificity of the answers provided 📜. This is followed by a detailed **Factual Checking** procedure, comprising five critical steps:

1. **Check for Made-Up Answers:** First, we look to see if any part of the answer seems made up or too vague.
2. **Break Down the Question:** We split the big question into smaller, easier questions to check each part more carefully.
3. **Find the Facts:** We use tools to find information from both general sources and specific data for each small question.
4. **Make the Information Clear:** We use smart models to summarize and organize the information so it's easy to understand.
5. **Check Each Part:** We use the models to look at the small answers and see if they match the information we found.
6. **Combine the Results:** Finally, we put all the small answers together to make a final judgment, making sure everything is accurate and complete.

## QA Task Hallucination Detection

To detect hallucinations in QA tasks, the following tools and parameters are used:

1. **qa\_relevance.py:** This script is used for Non-Fabrication Hallucination Checking, ensuring that the provided answers are not fabricated.
2. **qa\_query.py:** This script gathers queries and related knowledge for further analysis.

```
python qa_query.py --model Starling-LM-7B-alpha --form semantic --topk 2 --answer_type right --knowledge_type ground --query_selection None
```

Here are the parameters for detailed customization of `qa\_query.py`:

**model:** Specifies the language model to be used. The default is 'Starling-LM-7B-alpha'.

**form:** Determines the form of the data to be retrieved, which can be 'semantic' for unstructured data or structured forms.

**topk:** Sets the number of top results to retrieve from Wikipedia. The default is 2.

**answer\_type:** Defines the type of answer required, either 'right' (correct) or 'hallucinated' (fabricated).

**knowledge\_type:** Indicates the source of knowledge to be used, either 'ground' (off-the-shelf knowledge) or 'wiki' (retrieved knowledge from Wikipedia).

**query\_selection:** Specifies the index for the query formulation used. It can be 0 for specific queries, 1 for general queries, or None to use both.

### Final Judgments:

After collecting queries and relevant knowledge, final judgments are made using the `qa\_judge.py` script. This script evaluates the collected information and determines the accuracy and reliability of the answers.

In summary, the process involves checking for fabricated information, gathering relevant queries and knowledge, and then making final judgments based on the collected data. This structured approach ensures that the AI-generated content is reliable and factually accurate.

---

KnowHalu हे AI-निर्मित सामग्रीमधील काल्पनिक (hallucination) माहिती ओळखण्यास आणि व्यवस्थापित करण्यासाठी एक नाविन्यपूर्ण इष्टिकोन सादर करते, ज्यामुळे भाषा मॉडेल्सची विश्वसनीयता आणि तथ्यात्मक अचूकता वाढवता येते 🧠. आमचे कार्य एक व्यापक प्रक्रिया यावर आधारित आहे, ज्यामध्ये विविध टप्प्यांमधून तथ्य तपासणी केली जाते ✨.

हे प्रक्रिया या प्रकारे सुरु होते:

पहिल्या टप्प्यात बनावट माहिती तपासण (Non-Fabrication Hallucination Checking) केली जाते, ज्यामध्ये दिलेल्या उत्तरांच्या तपशीलावर आधारित काल्पनिक माहिती ओळखली जाते 🎭.

- बनावट माहिती ओळखणे:** उत्तराचा कोणताही भाग बनावट किंवा अस्पष्ट वाटतो का हे तपासणे.
- प्रश्नांचे विभाजन:** अवघड प्रश्नांना छोटे आणि सोपे भागांमध्ये विभाजित करणे, जेणेकरून त्यांची अधिक तपशीलाने पडताळणी करता येईल.
- माहिती मिळवणे:** प्रत्येक छोटे प्रश्नासाठी विविध स्रोतांमधून संबंधित माहिती शोधणे.
- सारांश आणि आयोजन:** प्रगत मॉडेल्सचा वापर करून ही माहिती स्पष्टपणे सारांशित आणि आयोजित करणे.
- उप-उत्तरांची पडताळणी:** मिळालेल्या माहितीच्या आधारावर प्रत्येक छोटे उत्तर सत्यापित करणे.
- मिळवलेली उत्तरे एकत्र करणे:** सर्व सत्यापित छोटे उत्तर एकत्र करून अंतिम निर्णय घेणे, यामुळे एकूण उत्तर अचूक आणि विश्वासार्ह बनते.

## प्रश्नोतरे (QA) कार्यातील काल्पनिक माहिती ओळखणे

प्रश्नोतरे (QA) कार्यातील काल्पनिक माहिती ओळखण्यासाठी खालील साधने आणि मापदंड वापरले जातात:

1. **qa\_relevance.py**: हा स्क्रिप्ट बनावट माहिती तपासण्यासाठी वापरला जातो, ज्यामुळे दिलेली उत्तरे बनावट नाहीत याची खात्री होते.

2. **qa\_query.py**: हा स्क्रिप्ट प्रश्न आणि संबंधित ज्ञान गोळा करण्यासाठी वापरला जातो.

`qa\_query.py` च्या सविस्तर सानुकूलनासाठी खालील मापदंड आहेत:

**model**: वापरायचे भाषेचे मॉडेल निर्दिष्ट करते. डीफॉल्ट म्हणजे 'Starling-LM-7B-alpha'.

**form**: डेटा कोणत्या स्वरूपात मिळवायचा हे ठरवते, जो 'semantic' (असंरचित डेटा) किंवा संरचित स्वरूप असू शकतो.

**topk**: विकिपीडियातून मिळवायच्या टॉप परिणामांची संख्या सेट करते. डीफॉल्ट म्हणजे 2.

**answer\_type**: आवश्यक उत्तराचा प्रकार निर्दिष्ट करते, जो 'right' (योग्य) किंवा 'hallucinated' (काल्पनिक) असू शकतो.

**knowledge\_type**: वापरायच्या ज्ञानाचा स्रोत दर्शवते, जो 'ground' (रेडीमेड ज्ञान) किंवा 'wiki' (विकिपीडियातून मिळवलेले ज्ञान) असू शकतो.

**query\_selection**: वापरलेल्या प्रश्न निर्माणाचे अनुक्रमणिका निर्दिष्ट करते. हे 0 (विशिष्ट प्रश्न), 1 (सामान्य प्रश्न) किंवा दोन्ही वापरण्यासाठी None असू शकते.

उदाहरण आदेश:

```
```bash
```

```
python qa_query.py --model Starling-LM-7B-alpha --form semantic --topk 2 --answer_type right --knowledge_type ground --query_selection None
```

```
```
```

अंतिम निर्णय:

प्रश्न आणि संबंधित ज्ञान गोळा केल्यानंतर, `qa\_judge.py` स्क्रिप्ट वापरून अंतिम निर्णय घेतले जातात. हा स्क्रिप्ट गोळा केलेली माहिती मूळ्यांकन करतो आणि उत्तरांची अचूकता आणि विश्वसनीयता निश्चित करतो.

सारांशात, ही प्रक्रिया बनावट माहिती तपासणे, संबंधित प्रश्न आणि ज्ञान गोळा करणे आणि गोळा केलेल्या डेटावर आधारित अंतिम निर्णय घेणे यांचा समावेश करते. ही संरचित पद्धत AI निर्मित सामग्री विश्वसनीय आणि तथ्यात्मक अचूक आहे याची खात्री देते.