Overview of the CLEF-2024 Eloquent Lab: Task 2 on HalluciGen

Detection and generation of hallucinations

Luise Dürlich[1], Evangelia Gogoulou[1], Liane Guillou[2], Joakim Nivre[1] and Shorouq Zahra[1]

[1]RISE Research Institutes of Sweden, Stockholm
[2]School of Informatics, University of Edinburgh

Abstract

In the HalluciGen task we aim to discover whether LLMs have an internal representation of hallucination. Specifically, we investigate whether LLMs can be used to both generate and detect hallucinated content. In the cross-model evaluation setting we take this a step further and explore the viability of using an LLM to evaluate output produced by another LLM. We include generation, detection, and cross-model evaluation steps for two scenarios: paraphrase and machine translation. Overall we find that performance of the baselines and submitted systems is highly variable, however initial results are promising and lessons learned from this year's task will provide a solid foundation for future iterations of the task. In particular, we highlight that human validation of generated output is ideally necessary to ensure the robustness of the cross-model evaluation results. We aim to address this challenge in future iterations of HalluciGen.

Keywords

Generative language models, Evaluation, Hallucinations

1. Introduction

Detecting hallucinations in LLM output may be difficult for humans in certain settings. For example, in the question answering scenario, an individual who asks an LLM a question about a domain with which they are unfamiliar might not be able to detect the presence of hallucinated content in the answer output by the model. In the cross-lingual setting the problem may become even more severe. For example, if the LLM is used to translate from or into a language that the human user does not comprehend well, they may be completely unable to identify hallucinations in the translation output. Models that humans will interact with should therefore be rigorously tested with respect to hallucination, prior to deployment.

In the HalluciGen task we aim to discover whether LLMs have an internal representation of hallucination – that is can they be used to both generate and detect hallucinated content? Taking this a step further, we also explore the viability of using LLMs in a cross-evaluation setting, where one LLM is used to evaluate the output of another [1, 2].

The first year of HalluciGen focused on developing models that are able to evaluate hallucination. Our task investigates the hallucination phenomenon in two downstream scenarios: **(i) Paraphrase Generation** (PG): given a source sentence, the model is instructed to produce an accurate paraphrase. For this scenario we include two languages: English and Swedish (en/sv); and **(ii) Machine Translation** (MT): given a sentence in a source language, the model is instructed to translate it into the target language. For this scenario we include two language pairs: English-German (en⇔de) and English-French (en⇔fr), for both translation directions. For each of the scenarios there are two steps:

- **Generation**: Given a source sentence, the model should generate two hypotheses, one that is a correct paraphrase/translation of the source ($hyp+$) and one that is a hallucinated paraphrase/-translation of the source ($hyp-$).

CLEF 2024: Conference and Labs of the Evaluation Forum, September 09–12, 2024, Grenoble, France
*The authors are listed in alphabetic order.

- **Detection**: Given a source sentence and two paraphrase/translation hypotheses ($hyp1$ and $hyp2$), the model should detect which of the two contains a hallucination.

As an additional challenge, we also perform the detection step in a *cross-model* setting, where the participant models perform the detection step on the model outputs from the generation step.

2. Datasets

For each of the two scenarios, i.e. paraphrase generation or machine translation, we construct a dataset with the following fields: a source sentence, a correct hypothesis of the source, a hallucinated hypothesis of the source, and the type of hallucination demonstrated in the hallucinated hypothesis. Our datasets include hallucinations of the following categories: addition, named-entity, number, conversion, date, gender, pronoun, antonym, tense, negation, and natural hallucinations. With the exception of tense and negation, the remainder of the hallucination types are identical to the type of translation errors identified in ACES [3]. All our datasets are available on Huggingface.[1] The process of dataset creation for each scenario is described below.

2.1. Machine Translation

For the translation scenario we leveraged ACES [3], a challenge set for evaluating the performance of Machine Translation (MT) metrics on a range of translation accuracy errors. Each example in ACES already follows the structure that we use in the HalluciGen task and ACES already contains errors for the en⇔fr and en⇔de language pairs for all but two of the phenomena we are interested in. For the tense and negation categories, which do not exist in ACES, we constructed examples from the PAWS-X dataset [4] of adversarial paraphrases.

For tense examples we filter the PAWS-X dataset to select English examples labelled as paraphrases, then we select each instance of sentence1 and use spaCy[2] to tokenise and part-of-speech tag the sentence. We then identify a verb and its tense using the Penn-treebank style tags output by the spaCy pipeline, and inflect it for a different tense using the pyinflect[3] python library. We change tense between past, present, and future (by injecting the token "will"). The original sentence1 forms the good translation, the perturbed version is the incorrect translation, and we pair the English sentence with the corresponding French/German translation in PAWS-X (which forms the source sentence). Negation examples are created by automatically extracting English paraphrase examples in PAWS-X that contain a negation and manually editing sentence1 to construct an incorrect translation e.g. by inserting an (extra) negation, or modifying the polarity of a sentence that already contains a negation. We consider lexical negation (e.g. the affixes "un" and "dis") and negation tokens (e.g. not, n't, never). Again, we pair the English target sentences with a corresponding French/German source sentence from PAWS-X.

From the combined set of ACES and the negation and tense examples, we selected 100 examples for each language direction for the test set and 10 examples for the trial set. Examples for the test set were selected in order to provide as close to a uniform selection across categories as possible. Note that due to the unbalanced coverage of examples in ACES, some categories are underrepresented or absent for some language directions.

2.2. Paraphrase

For the English paraphrase scenario, we sampled 138 examples from the SHROOM training data for the paraphrase generation subtask [5]. Each example consists of a source sentence accompanied with a machine-generated paraphrase hypothesis. The latter were generated by the SHROOM organisers

[1] https://huggingface.co/datasets/Eloquent/HalluciGen-PG
https://huggingface.co/datasets/Eloquent/HalluciGen-Translation
[2] https://spacy.io
[3] https://pypi.org/project/pyinflect/

using the PEGASUS model[4]. In order to increase the chance for hallucination, we prioritised examples with long contexts (minimum 140 tokens) that also include numbers.

In the Swedish paraphrase scenario, we used a subset of the SweParaphrase test data [6] and the Swedish part of the Finnish paraphrase corpus [7]. Each example consists of two sentences, together with a label reflecting the degree of their semantic similarity. After filtering only the sentence pairs with the highest degree of semantic similarity (that is label 5 in the Swedish Paraphrase dataset and label 4 in the Finnish paraphrase dataset), we sampled 139 examples and used two LLMs, Mixtral 7B [8] and GPT-SW3 6.7B [9], to generate a paraphrase hypothesis for the first sentence of each example. For these Swedish paraphrases, we observed cases where the generated paraphrase was in the wrong language, typically English, or a mix of languages when using Mixtral 7B. To obtain a large enough sample of reasonably good quality for annotation, we therefore chose to (1) translate output in English to Swedish using GPT3.5 and (2) generate multiple hypotheses – one by each of the LLMs – for some of these examples. In total, 46 of the sources had multiple annotations, while 56 sources only occur once in the entire dataset.

All datasets used for the paraphrase scenario are annotated in two steps. The first step is to decide if the generated hypothesis is a hallucination of the source, given the definition of the hallucination phenomenon in our task. If yes, then we mark the hypothesis as hallucination (H) and then choose a suitable hallucination type from the list of eleven hallucination categories in the HalluciGen dataset (addition, named-entity, etc.). If the hypothesis is marked as not hallucination (NH) then we construct a hallucination manually, based on one of the hallucination categories above. In the Swedish data, the cases with hypotheses in the wrong language or a mix of languages were considered too high effort to correct manually and were discarded from the final dataset.

Each of the resulting datasets per scenario and language was split into a trial and a test set. For English, 119 examples were selected for the test set and 16 examples for the trial set. The Swedish test set amounted to 119 examples in total (117 from SweParaphrase and 2 from the Finnish paraphrase corpus) and the trial set to 20 examples (19 from SweParaphrase and 1 from the Finnish corpus). The distribution of each dataset over the different hallucination types is presented in Table 1.

Table 1
Frequency of hallucination categories in the data

Language	Scenario	Data Split	Addition	Antonym	Date	Gender	Named Entity	Negation	Number	Pronoun	Tense	Conversion	Natural
en	PG	test	11	16	5	3	9	14	9	11	4	3	33
sv			42	11	–	3	15	12	9	1	5	1	20
en	PG	trial	2	2	1	1	2	1	1	2	–	1	3
sv			5	1	1	1	3	2	3	1	1	–	2
en-fr	MT	test	10	–	24	–	33	–	33	–	–	–	–
fr-en			9	13	4	12	12	12	13	–	12	13	–
en-de			10	16	14	–	15	–	13	16	–	–	16
de-en			10	10	7	11	10	10	10	–	10	11	11
en-fr	MT	trial	1	–	3	–	3	–	3	–	–	–	–
fr-en			1	1	1	1	1	2	1	–	1	1	–
en-de			1	1	2	–	1	–	3	1	–	–	1
de-en			1	1	1	1	1	1	1	–	1	1	1

[4]https://huggingface.co/tuner007/pegasus_paraphrase

3. Baseline Models

3.1. Paraphrase scenario

For the paraphrase scenario we use different models for the generation and detection steps. For generation, we use Mixtral-8x7B-Instruct-v0.1, the instructed variant of the Mixtral LLM [8][5] to generate $hyp+/hyp-$ hypotheses pairs for the English and Swedish test sets, and gpt-sw3-6.7b-v2 [9][6] as an additional baseline for Swedish.

For the detection step we use several models. The first is the Llama-2-7b-chat-hf[10] model from HuggingFace[7]. This model, although English-centric has been trained on smaller amounts of data for other languages, including Swedish. We use three prompts aimed at detecting which hypothesis a) is an incorrect paraphrase of the source, b) has a different meaning to the source, or c) is not supported by the source (see Table 9 in Appendix A). The second and third models are multilingual zero-shot Natural Language Inference (NLI) models, bge-m3-zeroshot-v2.0 [11] for both English and Swedish hallucination detection, and scandi-nli-large [12] [8] as an additional baseline for Swedish. These models classify a text into a number of custom defined classes; in our case, we choose the default classes "not_entailment" and "entailment" and infer the output label from the predicted scores for both classes. To determine which of the the two hypotheses ($hyp1/hyp2$) contains a hallucination, we predicted "entailment" and "not_entailment" class scores between the source sentence and each one of the hypotheses. We follow these conditions to infer the final label:

1. If one hypothesis has higher entailment whereas the other hypothesis has higher non-entailment, we choose the one with the higher non-entailment score.
2. If both $hyp1$ and $hyp2$ had a higher entailment score, we choose the one with the lowest entailment.
3. If both $hyp1$ and $hyp2$ had a higher non-entailment score, we choose the one with the highest non-entailment.

For both NLI models, the default configurations were used and each pair ($source{+}hyp1$ / $source{+}hyp2$). The models were used out of the box, as available on HuggingFace, and without any additional fine-tuning.

3.2. Machine translation scenario

For the translation scenario we again use the Llama2 [10] 7B-chat model from HuggingFace. We use this model as the baseline for the generation and detection steps. As stated in the previous section, while Llama2 is an English-centric model, it has been trained on (relatively) small amounts of data from other languages (including French and German) and is therefore able to perform cross-lingual tasks such as translation. Crucially, in addition to producing accurate translations it can also be prompted to produce incorrect translations in a zero-shot setting – something that we could not get MT-specific LLMs such as Tower [13] to do, perhaps because they have been optimised to output accurate translations. We note that there are many stronger LLMs, and our aim is not to provide an unbeatable baseline in the first year of HalluciGen.

For the generation step we split the problem into two parts, using separate prompts to produce the good and incorrect translations. For the good translation we simply prompt the model to translate from the source language to the target language. For the incorrect translation we use two different strategies; we prompt the model to a) produce an incorrect translation, and b) produce an incorrect translation and provide a list of possible phenomena that the incorrect translation could target. For

[5] https://huggingface.co/TheBloke/Mixtral-8x7B-Instruct-v0.1-GGUF
[6] https://huggingface.co/AI-Sweden-Models/gpt-sw3-6.7b-v2
[7] https://huggingface.co/meta-llama/Llama-2-7b-chat-hf
[8] https://huggingface.co/alexandrainst/scandi-nli-large

the detection step, similar to the paraphrase detection step, we have three different prompts aimed at detecting which hypothesis a) is an incorrect translation of the source, b) has a different meaning to the source, or c) is not supported by the source. We use the same prompts from the detection step in the cross-model evaluation step. For detailed of the exact prompts used, see Table 10 in Appendix A. Note that we experimented with explicitly including the term "hallucination" as part of the prompt instructions, but this was unsuccessful.

We used the default Llama2-7B-chat model parameters, unless otherwise stated. For generation (translation only) we want to encourage creativity for translation so we set temperature to 0.9; top_k=10, num_return_sequences=1, and max_length=200. For detection (for both translation and paraphrase) we want to encourage deterministic behaviour so we set temperature to 0.1 and top_k=1; as our prompts are longer than for the generation step we set max_length=400 (to allow for longer inputs).

In addition to Llama2, we again employed BGE-M3-ZEROSHOT-V2.0 [11] to create detection step baselines for all language pairs and directions. This uses the same model and process detailed in Paraphrase scenario section. Although the model performs better on English input, it is still suitable for multilingual tasks. While it was recommended to translate input sentences into English rather than having them in multiple languages (as a way to improve performance), no additional translation was performed on either the source sentences nor the hypotheses pair ($hyp1/hyp2$); this means that the NLI model receives two sentence in two different languages as input (one in English, and one in either French or German) in both directions.

4. Participant Submissions

In total, we received outputs from 10 systems submitted by 3 different groups which included varying numbers of participants. Table 2 provides an overview of the submitted systems. Participant group 1 (Bui et al.) [14] submitted systems for all steps and all languages for both the paraphrase and translation scenarios. They applied zero-shot prompting for a range of pre-trained LLMs, and ensembled combinations of these models to produce majority voting systems. Participant group 3 (Siino & Tinnirello) [15] submitted systems for the detection step of the paraphrase scenario only. They used MISTRAL-7B-INSTRUCT-V0.2 with few-shot prompting, providing the complete set of examples (either English or Swedish depending on the language in focus) from the trial data set as part of the prompt. Participant group 2 (Abburi) submitted systems for the detection step for both the paraphrase and translation scenarios. Unfortunately, as they did not submit a paper to CLEF 2024, we know little about their system other than it uses majority voting across multiple fine-tuned LLMs.

5. Evaluation Methodology

5.1. Detection Step

For the detection step, the submitted systems are evaluated with respect to the human-annotated labels, using the following metrics: accuracy, precision, recall and F1 score. We use F1 as the primary metric for comparison between different systems. Examples were classified as incorrect in cases when the evaluated system produced no label or a label outside the allowed categories ($hyp1/hyp2$).

5.1.1. Generation Step

We use the NLI task as a proxy for evaluating the quality of the correct and hallucinated hypothesis $hyp+$,$hyp-$ generated by the participant models. More specifically, the NLI model BGE-M3-ZEROSHOT-V2.0 [11], that also serves as a baseline for the detection step, is now used to predict "entailment" vs "not_entailment" scores. The rationale behind this is as follows: one way to determine whether or not a system is able to create appropriate pairs of hypotheses is to measure the textual entailment between each pair and the source sentence. We assume that a successful paraphrase of a sentence textually entails the source sentence; whereas a hallucination does not. If $hyp+$ is predicted as having

Table 2
Participant systems by task and scenario (PG and MT), including the languages or language pairs for which output was submitted. The double-direction arrows "⇔" indicates participant submissions for a language pair in both directions

LLM System	Detection	Generation	Cross-Model evaluation
Participant Group 1 (Bui et al.)			
google/gemma-7b-it	PG (en/sv) MT (en⇔de) MT (en⇔fr)	PG (en/sv) MT (en⇔de) MT (en⇔fr)	PG (en/sv) MT (en⇔de) MT (en⇔fr)
gpt-3.5-turbo	PG (en/sv) MT (en⇔de) MT (en⇔fr)	PG (en/sv) MT (en⇔de) MT (en⇔fr)	PG (en/sv) MT (en⇔de) MT (en⇔fr)
gpt-4	PG (en/sv) MT (en⇔de) MT (en⇔fr)	-	MT (en⇔de) MT (en⇔fr)
gpt-4-turbo	PG (en/sv)	-	PG (en/sv)
meta-llama/Meta-Llama-3-8B-Instruct	PG (en/sv) MT (en⇔de) MT (en⇔fr)	PG (en) MT (en⇔de) MT (en⇔fr)	PG (en/sv) MT (en⇔de) MT (en⇔fr)
meta-llama/Meta-Llama-3-8B	PG (en/sv)	-	-
Majority vote (A) on: google/gemma-7b-it meta-llama/Meta-Llama-3-8B-Instruct gpt-3.5-turbo gpt-4-turbo	PG (en/sv) MT (en⇔de) MT (en⇔fr)	-	PG (en/sv)
Majority vote (B) on: google/gemma-7b-it meta-llama/Meta-Llama-3-8B-Instruct gpt-3.5-turbo gpt-4	-	-	PG (en/sv) MT (en⇔de) MT (en⇔fr)
Participant Group 2 (Abburi)			
Majority voting of finetuned LLMs	PG (en/sv) MT (en⇔de) MT (en⇔fr)	-	-
Participant Group 3 (Siino & Tinnirello)			
TheBloke/Mistral-7B-Instruct-v0.2-GGUF	PG (en/sv)	-	-

higher "entailment", it is assigned a score of 1, otherwise 0, and if a $hyp-$ is is predicted as having higher "not_entailment", it is assigned a score of 1, otherwise 0. To validate the use of the NLI model for evaluating the model outputs for the generation step, we test the NLI model BGE-M3-ZEROSHOT-V2.0 as a baseline for the detection step in both scenarios. These are the scores highlighted in grey in Tables 3 and 7. We observe that the NLI model competes with (or even surpasses) the participant models on the detection task. This allows us to use it for evaluating the model outputs for the generation step.

5.1.2. Cross-model evaluation

For the cross-model evaluation, the system performance is measured with respect to the output of the generator model, using the same metrics as in the detection step. In addition, Matthew's correlation coefficient (mcc) and Cohen's kappa are used to measure the agreement between the different evaluators.

6. Results

6.1. Paraphrase Scenario

Tables 3 - 5 present the results of the participant models and the baselines for the three steps of the paraphrase scenario. Starting from the detection step, we observe that the NLI baseline BASELINE-BGE-M3-ZEROSHOT-V2.0 exhibits very strong performance. The difference with the participant models is even

Table 3
Detection step results for the paraphrase scenario. Results for the NLI model BGE-M3-ZEROSHOT-V2.0 (highlighted in grey) are included for the purpose of validating the NLI model as an evaluation method for the generation step.

Detection: Paraphrase			
LLM system	F1	**LLM system**	F1
English		**Swedish**	
gemma-7b-it	0.49	gemma-7b-it	0.11
gemma-7b-it v1	0.71	gemma-7b-it v1	0.52
gpt-3.5-turbo	0.68	gpt-3.5-turbo	0.60
gpt-3.5-turbo v1	0.73	gpt-3.5-turbo v1	0.70
gpt-4-turbo	**0.91**	gpt-4-turbo	0.81
Meta-Llama-3-8B-Instruct	0.80	Meta-Llama-3-8B-Instruct	0.59
Meta-Llama-3-8B	0.69	Meta-Llama-3-8B	0.48
Majority vote A (Bui et al.)	0.85	Majority vote (Abburi)	0.79
Majority vote (Abburi)	0.90	Majority vote A (Bui et al.)	0.66
Mistral-7B-Instruct-v0.2	0.72	Mistral-7B-Instruct-v0.2	0.75
Baselines			
baseline-bge-m3-zeroshot-v2.0	0.90	baseline-bge-m3-zeroshot-v2.0	**0.92**
baseline-llama2-meaning-detection	0.44	baseline-llama2-meaning-detection	0.60
baseline-llama2-not-supported-detection	0.35	baseline-llama2-not-supported-detection	0.56
baseline-llama2-paraphrase-detection	0.35	baseline-llama2-paraphrase-detection	0.59
		baseline-sv_scandi-nli-large	**0.92**

Table 4
Generation results for the paraphrase scenario. $hyp+$, $hyp-$ refer to the accuracy of the NLI model on predicting that $hyp+$ is entailed and $hyp-$ is not entailed correspondingly.

Generation: Paraphrase					
LLM system	$hyp+$	$hyp-$	**LLM system**	$hyp+$	$hyp-$
English			**Swedish**		
gemma-7b-it v1	0.82	0.89	gemma-7b-it v1	0.35	**0.93**
gemma-7b-it v2	0.85	0.90	gemma-7b-it v2	0.61	0.69
gpt-3.5-turbo	**0.98**	0.80	gpt-3.5-turbo	**0.90**	**0.93**
Meta-Llama-3-8B-Instruct	0.88	**0.98**			
Baselines					
baseline-mixtral-8x7b-instruct	0.92	0.74	baseline-gpt-sw3-6.7b-v2	0.64	0.50
			baseline-mixtral-8x7b-instruct	0.84	0.35

Table 5
Cross-model step results for the paraphrase scenario.

Cross-model evaluation: Paraphrase					
LLM system	F1	Avg Kappa	**LLM system**	F1	Avg Kappa
English			**Swedish**		
gemma-7b-it v1	0.77	0.61	gemma-7b-it v1	0.48	0.19
gpt-3.54-turbo v2	0.88	0.77	gpt-3.54-turbo v2	0.68	0.48
Meta-Llama-3-8B-Instruct	0.92	0.74	Meta-Llama-3-8B-Instruct	0.70	0.50
Majority vote A (Bui et al.)	0.92	**0.81**	Majority vote A (Bui et al.)	**0.76**	**0.59**
gpt-4-turbo v2	**0.93**	0.75	gpt-4-turbo v2	0.74	0.41

more noticable for the Swedish dataset, where the best performing participant model, GPT-4-TURBO lies over 10 points behind the NLI baseline in terms of F1 score. This is almost expected since none of the participant models has been (intentionally) trained on Swedish data. For the English paraphrase, GPT-4-TURBO and the MAJORITY VOTE (ABBURI) models perform on the same level as the baseline on the task of hallucination detection.

For the generation step, GPT-3-5-TURBO produces overall the best quality positive and negative hypotheses in both English and Swedish, according to the NLI model. Notably larger difference between the $hyp+$ and $hyp-$ scores of that model is observed in English, in comparison with Swedish. In addition, GEMMA-7B-IT V1 stands out for generating $hyp-$ hypotheses with considerably better quality

than $hyp+$ hypotheses, according to the NLI model.

From the results of the cross-model evaluation in Table 5 we observe that the MAJORITY VOTE A (BUI ET AL.) exhibits the best overall performance in detecting hallucinations in machine-generated hypotheses in English and Swedish, with respect to both the generator output and the other evaluator models.

Table 6
Generation step results for the translation scenario

Generation: Translation								
	en-fr		fr-en		en-de		de-en	
LLM system	$hyp+$	$hyp-$	$hyp+$	$hyp-$	$hyp+$	$hyp-$	$hyp+$	$hyp-$
Meta-Llama-3-8B-Instruct prompt1 (Bui et al.)	0.77	0.81	0.82	0.88	0.84	**0.84**	0.84	0.85
Meta-Llama-3-8B-Instruct prompt2 (Bui et al.)	0.81	**0.86**	0.81	**0.96**	0.84	0.68	0.85	0.62
gemma-7b-it (Bui et al.)	0.80	0.49	0.73	0.57	**0.85**	0.42	0.70	0.54
gpt-3.5-turbo (Bui et al.)	0.88	0.91	0.86	0.90	0.81	**0.84**	0.86	**0.95**
Llama-2-7b-chat-hf general-prompt	**0.93**	0.08	**1.00**	0.03	**0.85**	0.19	**0.98**	0.03
Llama-2-7b-chat-hf phenomena-mentions-prompt	0.92	0.23	0.97	0.08	**0.85**	0.33	**0.98**	0.06

Table 7
Detection step results for the translation scenario. Results for the NLI model BGE-M3-ZEROSHOT-V2.0 (highlighted in grey) are included for the purpose of validating the NLI model as an evaluation method for the generation step.

Detection: Translation				
	F1			
LLM system	**en-fr**	**fr-en**	**en-de**	**de-en**
Meta-Llama-3-8B-Instruct final (Bui et al.)	0.51	0.63	0.47	0.67
Meta-Llama-3-8B-Instruct new-prompt-final (Bui et al.)	0.65	0.60	0.69	0.70
gemma-7b-it (Bui et al.)	0.66	0.61	0.59	0.58
gemma-7b-it final (Bui et al.)	0.60	0.46	0.54	0.53
gpt-3.5-turbo prompt1 (Bui et al.)	0.74	0.75	0.67	0.80
gpt-3.5-turbo prompt2 (Bui et al.)	0.76	0.82	0.80	0.83
gpt-4 prompt1 (Bui et al.)	**0.90**	0.87	**0.86**	**0.93**
gpt-4 prompt2 (Bui et al.)	0.79	**0.89**	0.79	0.83
Majority vote A (Bui et al.)	0.83	0.84	0.81	0.85
Majority vote (Abburi)	0.85	0.87	0.85	0.89
bge-m3-zeroshot-v2.0	0.82	0.88	0.73	0.78
Llama-2-7b-chat-hf general-prompt	0.47	0.50	0.48	0.50
Llama-2-7b-chat-hf meaning-prompt	0.50	0.44	0.36	0.36
Llama-2-7b-chat-hf supported-prompt	0.24	0.35	0.41	0.50

Table 8
Cross-model evaluation step results for the translation scenario

Cross-model Evaluation: Translation								
	wrt. generator output (F1)				wrt. other evaluators (K)			
LLM system	**en-fr**	**fr-en**	**en-de**	**de-en**	**en-fr**	**fr-en**	**en-de**	**de-en**
Meta-Llama-3-8B-Instruct final (Bui et al.)	0.65	0.68	0.52	0.51	0.43	0.45	0.27	0.33
gemma-7b-it final (Bui et al.)	0.57	0.53	0.53	0.55	0.23	0.13	0.15	0.17
gpt-3.5-turbo (Bui et al.)	0.77	0.75	**0.75**	0.71	0.57	0.55	0.50	0.54
gpt-4 (Bui et al.)	0.76	0.75	0.73	0.71	0.59	0.58	0.52	0.53
Majority vote B (Bui et al.)	**0.78**	**0.79**	0.74	**0.73**	**0.65**	**0.62**	**0.58**	**0.59**

6.2. Machine translation Scenario

Tables 6 - 8 contain the results for the translation scenario. For the generation step (Table 6) we observe that performance of LLAMA-3-8B-INSTRUCT and GPT-3.5-TURBO participant systems is generally good: the average "entailment" scores for $hyp+$ and "not_entailment" scores for $hyp-$ suggest that the models are generally consistent in their ability to generate hypotheses that are entailed by the reference ($hyp+$) and that contradict the reference ($hyp-$). The two LLAMA-2-7B-CHAT baselines and, to a lesser degree, the GEMMA-7B-IT participant system exhibit stronger performance for the generation of $hyp+$

examples than $hyp-$ examples. In particular, the Llama-2-7b-chat baselines outperform the participant systems for the task of generating $hyp+$ examples. We conjecture that this may be a result of using separate prompts to generate $hyp+$ and $hyp-$; by focusing the prompt for generating $hyp+$ examples on generating a "good" translation of the source we may focus the model on the translation task, for which it was likely fine-tuned. Conversely, the baseline performance for generating $hyp-$ examples is very low, but confidence in the ability of LLMs to perform this task is buoyed by the performance of the participant systems. Note that these results are based on automatic metrics; for a complete evaluation we propose that the generated output be verified by human annotators, which we leave to future work.

For the detection step, all participant systems outperformed the Llama-2-7b-chat baselines (one model; three different prompts). The stronger bge-m3-zeroshot-v2.0 baseline, is outperformed by a number of participant systems for all language pairs. Overall, gpt-4 prompt1 is the strongest-performing participant system, with the highest F1 score for three out of four language pairs. The majority voting strategies of Bui et al. [14] and Abburi also perform strongly.

For the cross-model evaluation step, from which we exclude the baselines, we find that the majority voting strategy of Bui et al. [14] works well, with strong F1 performance on detection based on the examples generated by the models in the generation step, and also has the highest agreement (measured using Cohen's Kappa) with the other evaluator models.

7. Conclusion and Future Work

In the HalluciGen task we explored the use of LLMs in generating and detecting hallucinations in paraphrase and translation tasks. We find that performance of the participant and baseline systems is highly variable, but results from this year's lab are promising and will provide a solid foundation for future iterations of the task. We highlight that all three steps (generation, detection, and cross-model evaluation) have been evaluated automatically, and therefore caution the reader against drawing any conclusions regarding which models, prompts, or methods may be "best" based solely on the results in this paper. In the case of the generation step in particular, human validation of the generated output is ideally necessary to ensure the robustness of the cross-model evaluation results. We aim to address this challenge in future iterations of HalluciGen.

Acknowledgments

This lab has been partially supported by the Swedish Research Council (grant number 2022-02909) and by UK Research and Innovation (UKRI) under the UK government's Horizon Europe funding guarantee [grant number 10039436 (Utter)].

References

[1] P. Manakul, A. Liusie, M. J. F. Gales, Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models, arXiv preprint: 2303.08896 (2023).

[2] W. Saunders, C. Yeh, J. Wu, S. Bills, L. Ouyang, J. Ward, J. Leike, Self-critiquing models for assisting human evaluators, arXiv preprint: 2206.05802 (2022).

[3] C. Amrhein, N. Moghe, L. Guillou, ACES: Translation accuracy challenge sets for evaluating machine translation metrics, in: Proceedings of the Seventh Conference on Machine Translation (WMT), Association for Computational Linguistics, ????, pp. 479–513. URL: https://aclanthology.org/2022.wmt-1.44.

[4] Y. Yang, Y. Zhang, C. Tar, J. Baldridge, PAWS-X: A cross-lingual adversarial dataset for paraphrase identification, in: K. Inui, J. Jiang, V. Ng, X. Wan (Eds.), Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong

Kong, China, 2019, pp. 3687–3692. URL: https://aclanthology.org/D19-1382. doi:10.18653/v1/D19-1382.
[5] T. Mickus, E. Zosa, R. Vázquez, T. Vahtola, J. Tiedemann, V. Segonne, A. Raganato, M. Apidianaki, Semeval-2024 shared task 6: Shroom, a shared-task on hallucinations and related observable overgeneration mistakes, arXiv preprint arXiv:2403.07726 (2024).
[6] A. Berdicevskis, G. Bouma, R. Kurtz, F. Morger, J. Öhman, Y. Adesam, L. Borin, D. Dannélls, M. Forsberg, T. Isbister, A. Lindahl, M. Malmsten, F. Rekathati, M. Sahlgren, E. Volodina, L. Börjeson, S. Hengchen, N. Tahmasebi, Superlim: A Swedish language understanding evaluation benchmark, Association for Computational Linguistics, Singapore, 2023, pp. 8137–8153.
[7] J. Kanerva, F. Ginter, L.-H. Chang, I. Rastas, V. Skantsi, J. Kilpeläinen, H.-M. Kupari, J. Saarni, M. Sevón, O. Tarkka, Finnish paraphrase corpus, in: Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa), Linköping University Electronic Press, Sweden, Reykjavik, Iceland (Online), 2021, pp. 288–298.
[8] A. Q. Jiang, A. Sablayrolles, A. Roux, A. Mensch, B. Savary, C. Bamford, D. S. Chaplot, D. d. l. Casas, E. B. Hanna, F. Bressand, et al., Mixtral of experts, arXiv preprint: 2401.04088 (2024).
[9] A. Ekgren, A. Cuba Gyllensten, F. Stollenwerk, J. Öhman, T. Isbister, E. Gogoulou, F. Carlsson, J. Casademont, M. Sahlgren, GPT-SW3: An autoregressive language model for the Scandinavian languages, in: Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), Torino, Italia, 2024.
[10] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, T. Scialom, Llama 2: Open foundation and fine-tuned chat models, 2023. arXiv:2307.09288.
[11] M. Laurer, W. van Atteveldt, A. Casas, K. Welbers, Building Efficient Universal Classifiers with Natural Language Inference, 2023. URL: http://arxiv.org/abs/2312.17543. doi:10.48550/arXiv.2312.17543, arXiv:2312.17543 [cs].
[12] D. S. Nielsen, Scandinli: Natural language inference for the scandinavian languages, https://github.com/alexandrainst/ScandiNLI, 2022. URL: https://aclanthology.org/D19-1382.
[13] D. M. Alves, J. Pombal, N. M. Guerreiro, P. H. Martins, J. Alves, A. Farajian, B. Peters, R. Rei, P. Fernandes, S. Agrawal, P. Colombo, J. G. C. de Souza, A. F. T. Martins, Tower: An open multilingual large language model for translation-related tasks, 2024. arXiv:2402.17733.
[14] A. T. Bui, S. F. Brech, N. Hußfeldt, T. Jennert, M. Ullrich, T. Breuer, N. Nikzad, P. Schaer, The two sides of the coin: Hallucination generation and detection with evaluators for llms, in: G. Faggioli, N. Ferro, P. Galuščáková, A. García Seco de Herrera (Eds.), Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum, 2024.
[15] M. Siino, I. Tinnirello, Gpt hallucination detection through prompt engineering, in: G. Faggioli, N. Ferro, P. Galuščáková, A. García Seco de Herrera (Eds.), Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum, 2024.

A. Task 2 - Baseline System Prompts

The prompts used for the paraphrase and translation baseline LLM systems are provided in Tables 9 and 10 respectively.

Table 9
Prompts for Paraphrase baseline models.In the **generation step**, the model is instructed to generate a pair of hypotheses (sometimes explicitly named "hyp+" or "hyp-") where one is supported by the source sentence and the other is not. In the **detection step**, the model is instructed to identify which of the two hypotheses, hypothesis1 (hyp1) or hypothesis2 (hyp2) contains the hallucinated content, given the source sentence.

Model	Prompt
Paraphrase: Generation Step	
gpt-sw3-6.7b-v2	Generera en parafras hyp+ som stöds av src och en andra parafras hyp- som inte stöds av src.
mixtral-8x7b-instruct	**Prompt for English:** Given the src below, generate a paraphrase hypothesis hyp+ that is supported by src and a paraphrase hypothesis hyp- that is not supported by src. **Prompt for Swedish:** Generera en parafras hyp+ som stöds av src och en andra parafras hyp- som inte stöds av src.
Paraphrase: Detection Step	
Llama2-7B-general-prompt	Which hypothesis is an incorrect paraphrase of the source: hypothesis1 or hypothesis2? source: <source> hypothesis1: <hyp1> hypothesis2: <hyp2> Acceptable answers: 'hypothesis1', 'hypothesis2' Answer:
Llama2-7B-meaning-prompt	Given the source which hypothesis contains content which is not present in the source, or has a different meaning to the source: hypothesis1 or hypothesis2? source: <source> hypothesis1: <hyp1> hypothesis2: <hyp2> Acceptable answers: 'hypothesis1', 'hypothesis2' Answer:
Llama2-7B-support-prompt	Which hypothesis is not supported by the source: hypothesis1 or hypothesis2? source: <source> hypothesis1: <hyp1> hypothesis2: <hyp2> Acceptable answers: 'hypothesis1', 'hypothesis2' Answer:

Table 10
Prompts for Translation baseline models. In the **generation step** the model is instructed to produce translations of src_sentence, a source language (src_lang) text into the target language (tgt_lang). In the **detection step** the model is instructed to identify which of the two hypotheses, hypothesis1 ($hyp1$) or hypothesis2 ($hyp2$) contains the hallucinated content, given the source sentence.

Model	Prompt
Translation: Generation Step	
Llama2-7B (good translation)	Translate the following <src_lang> text into <tgt_lang> Text: <src_sentence> <tgt_lang>:
Llama2-7B-general-prompt (incorrect translation)	Translate the following <src_lang> text incorrectly into <tgt_lang> Text: <src_sentence> <tgt_lang>:
Llama2-7B-mentions-prompt (incorrect translation)	Translate the following <src_lang> text incorrectly into <tgt_lang> and change its meaning, for example by inserting a word, changing the tense of the text, negating the text, or replacing a date, number, named entity, or pronoun. Text: <src_sentence> <tgt_lang>:
Translation: Detection Step	
Llama2-7B-general-prompt	Which <tgt_lang> hypothesis is an incorrect translation of the <src_lang> source: hypothesis1 or hypothesis2? source: <src> hypothesis1: <hyp1> hypothesis2: <hyp2> Acceptable answers: 'hypothesis1', 'hypothesis2' Answer:
Llama2-7B-meaning-prompt	Given the <src_lang> source which <tgt_lang> hypothesis contains content which is not present in the source, or has a different meaning to the source: hypothesis1 or hypothesis2? source: <source> hypothesis1: <hyp1> hypothesis2: <hyp2> Acceptable answers: 'hypothesis1', 'hypothesis2' Answer:
Llama2-7B-support-prompt	Which hypothesis is not supported by the source: hypothesis1 or hypothesis2? source: <source> hypothesis1: <hyp1> hypothesis2: <hyp2> Acceptable answers: 'hypothesis1', 'hypothesis2' Answer: