SHROOM-INDElab at SemEval-2024 Task 6: Zero- and Few-Shot LLM-Based Classification for Hallucination Detection

Bradley P. Allen, Fina Polat and **Paul Groth**
University of Amsterdam
Amsterdam, NL
{b.p.allen, f.yilmazpolat, p.t.groth}@uva.nl

Abstract

We describe the University of Amsterdam Intelligent Data Engineering Lab team's entry for the SemEval-2024 Task 6 competition. The SHROOM-INDElab system builds on previous work on using prompt programming and in-context learning with large language models (LLMs) to build classifiers for hallucination detection, and extends that work through the incorporation of context-specific definition of task, role, and target concept, and automated generation of examples for use in a few-shot prompting approach. The resulting system achieved fourth-best and sixth-best performance in the model-agnostic track and model-aware tracks for Task 6, respectively, and evaluation using the validation sets showed that the system's classification decisions were consistent with those of the crowd-sourced human labellers. We further found that a zero-shot approach provided better accuracy than a few-shot approach using automatically generated examples. Code for the system described in this paper is available on Github[1].

1 Introduction

Prompt engineering of large language models (LLMs) (Liu et al., 2023) has recently emerged as a viable approach to the automation of a wide range of natural language processing tasks. Recent work (Allen, 2023) has focused on the development of zero-shot chain-of-thought (Wei et al., 2022; Kojima et al., 2022) classifiers, where hallucination in generated rationales is a concern. Hallucination detection (Ji et al., 2023; Huang et al., 2023) is a way to determine whether the outputs of such systems are sensible, factually correct and faithful to the provided input. The SemEval-2024 Task 6 (Mickus et al., 2024) allows us to evaluate whether and how applying techniques we have developed in the above mentioned work and with related work on knowledge extraction (Polat et al., 2024) using zero- and few-shot classification can provide a means of addressing this concern. Previous systems that perform prompt engineering of LLMs as a means to implement hallucination detection include SelfCheckGPT (Manakul et al., 2023) and ChainPoll (Friel and Sanyal, 2023).

2 Data and Task

The challenge provides a dataset consisting of data points containing: the specific task that a given language model is to perform; an input given to the language model on which to perform that task; a target that is an example of an acceptable output, and the output produced by the language model. Table 1 shows an example of such a data point.

Task	Definition Modeling
Input text	"The Dutch would sometimes <define> inundate </define> the land to hinder the Spanish army ."
Target text	"To cover with large amounts of water; to flood."
Generated text	"(transitive) To fill with water."

Table 1: Example data point from the unlabeled training dataset for the model-agnostic task.

Hallucination detection is framed as a binary classification task, where the classifier assigns either 'Hallucination' or 'Not Hallucination' labels with associated probability estimates to data points. Classifier performance is evaluated by comparing these assignments and probabilities to human judgments and their probability estimates, using accuracy and Spearman's correlation coefficient (ρ) for assessment. Around 200 crowd-sourced human labellers each labeled about 20 data points. The competition features two tracks: model-agnostic, which uses the basic setup, and model-aware, adding a field for the Hugging Face model identifier of the model generating the text for each data point. Each track provides an unlabeled training dataset and labeled validation and test datasets.

[1] https://www.github.com/bradleypallen/shroom/

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 839–844
June 20-21, 2024 ©2024 Association for Computational Linguistics

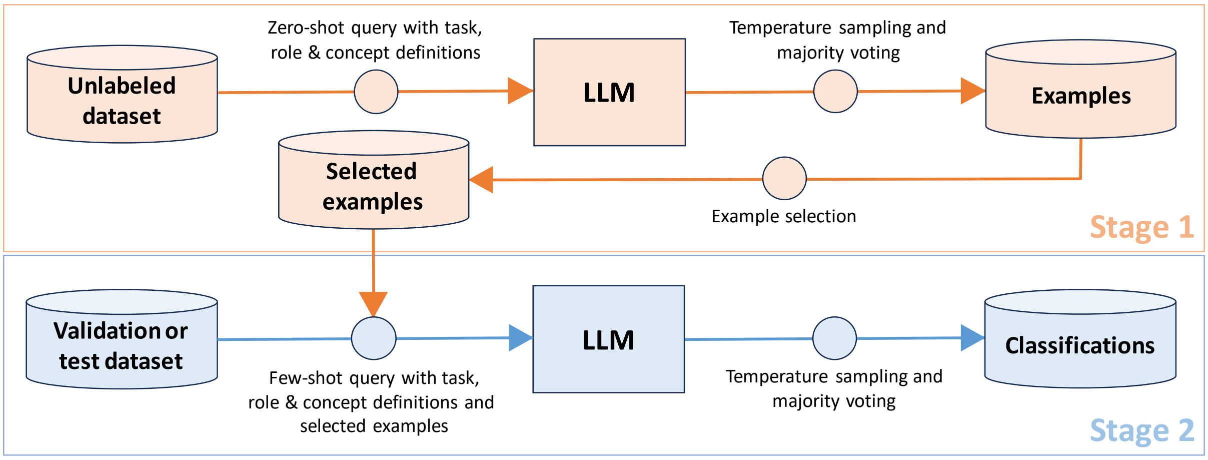

Unlabeled dataset
Zero-shot query with task, role & concept definitions
LLM
Temperature sampling and majority voting
Examples
Selected examples
Example selection
Stage 1
Validation or test dataset
Few-shot query with task, role & concept definitions and selected examples
LLM
Temperature sampling and majority voting
Classifications
Stage 2

Figure 1: SHROOM-INDElab system workflow.

3 Approach

Our submission for the SHROOM task is a system that defines classifiers for hallucination detection using prompt engineering of an LLM. Figure 1 shows the two-stage workflow used to produce the classifier and evaluate it using the SHROOM datasets.

In Stage 1, we use in-context learning where we ask the LLM to perform the classification according to provided task, role, and concept definition in a zero-shot manner without providing any examples. These classified data points provide examples for a few-shot classifier used in Stage 2. We now proceed to describe the query design and processing steps in the workflow.

3.1 Zero- and few-shot query design

Figure 2 provides an example of the query used to prompt an LLM to produce a classification. The basic prompt template consists of instructions on how to evaluate the generated text according to a hallucination concept definition to answer the question if the generated text is a hallucination or not. Specific guidance is provided such that the form of the answer is in the labels needed to compare directly to the label test data.

The task associated with the data point determines the context for generating both the zero shot and the few shot query based on the prompt template, as illustrated in Figure 2. For the zero-shot query, no examples are included.

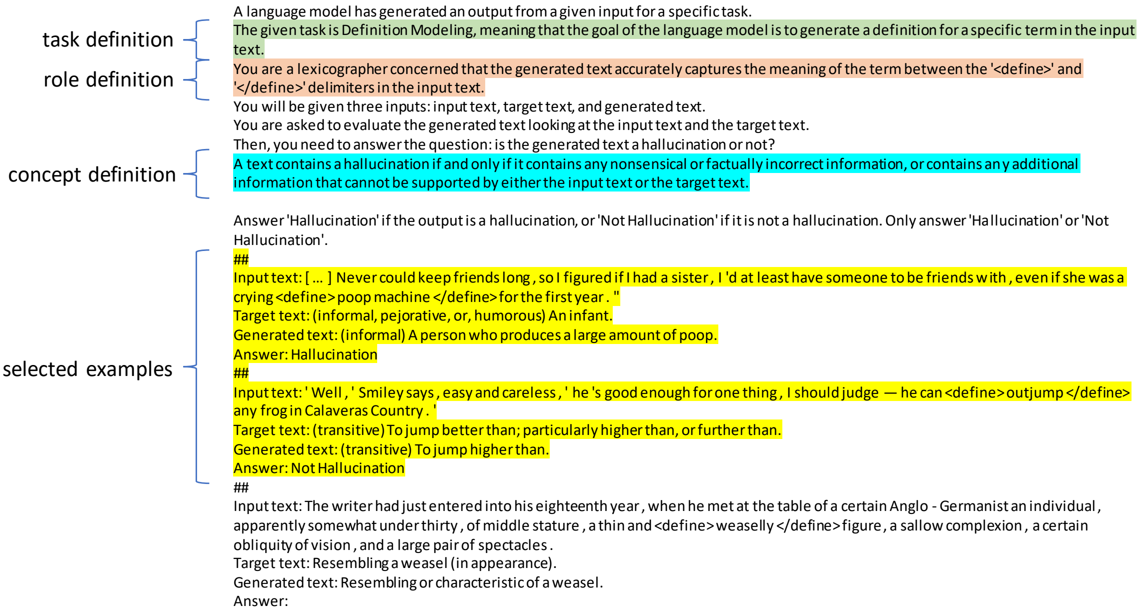
The elements involved in instantiating the template given a data point include the task definition performed by another LLM to produce the generated text, a role definition that we assign the classifier to perform, and the concept definition that frames hallucination phenomena and criteria to consider an output as hallucination. The use of role play with LLMs is described by (Shanahan et al., 2023) and its use in the context of zero-shot reasoning is described in (Kong et al., 2023). The role definition describes a persona that the LLM is instructed to assume in the context of making a classification decision. For example, for the Definition Modeling task, we instruct the LLM to assume the persona of a lexicographer. The task and role definitions for each task are shown in Table 2. We also provide a single concept definition for the notion of hallucination that is held constant across all of the tasks.

Task	Task definition	Role definition
Definition Modeling (DM)	The given task is Definition Modeling, meaning that the goal of the language model is to generate a definition for a specific term in the input text.	You are a lexicographer concerned that the generated text accurately captures the meaning of the term between the '<define>' and '</define>' delimiters in the input text.
Paraphrase Generation (PG)	The given task is Paraphrase Generation, meaning that the goal of the language model is to generate a paraphrase of the input text.	You are an author concerned that the generated text is an accurate paraphrase that does not distort the meaning of the input text.
Machine Translation (MT)	The given task is Machine Translation, meaning that the goal of the language model is to generate a natural language translation of the input text.	You are a translator concerned that the generated text is a good and accurate translation of the input text.
Text Simplification (TS)	The given task is Text Simplification, meaning that the goal of the language model is to generate a simplified version of the input text.	You are an editor concerned that the generated text is short, simple, and has the same meaning as the input text.

Table 2: Task and role definitions used for in-context learning.

3.2 Temperature sampling and majority voting

Part of the task involves producing an estimate of the probability that a data point exhibits hallucination. In the SHROOM-INDELab system, the estimated probability is calculated by performing temperature sampling (Ackley et al., 1985), query-



task definition
role definition
concept definition
selected examples

```
A language model has generated an output from a given input for a specific task.
The given task is Definition Modeling, meaning that the goal of the language model is to generate a definition for a specific term in the input
text.
You are a lexicographer concerned that the generated text accurately captures the meaning of the term between the '<define>' and
'</define>' delimiters in the input text.
You will be given three inputs: input text, target text, and generated text.
You are asked to evaluate the generated text looking at the input text and the target text.
Then, you need to answer the question: is the generated text a hallucination or not?
A text contains a hallucination if and only if it contains any nonsensical or factually incorrect information, or contains any additional
information that cannot be supported by either the input text or the target text.

Answer 'Hallucination' if the output is a hallucination, or 'Not Hallucination' if it is not a hallucination. Only answer 'Hallucination' or 'Not
Hallucination'.
##
Input text: [ ... ] Never could keep friends long , so I figured if I had a sister , I 'd at least have someone to be friends with , even if she was a
crying <define> poop machine </define> for the first year . "
Target text: (informal, pejorative, or, humorous) An infant.
Generated text: (informal) A person who produces a large amount of poop.
Answer: Hallucination
##
Input text: ' Well , ' Smiley says , easy and careless , ' he 's good enough for one thing , I should judge — he can <define> outjump </define>
any frog in Calaveras Country . '
Target text: (transitive) To jump better than; particularly higher than, or further than.
Generated text: (transitive) To jump higher than.
Answer: Not Hallucination
##
Input text: The writer had just entered into his eighteenth year , when he met at the table of a certain Anglo - Germanist an individual ,
apparently somewhat under thirty , of middle stature , a thin and <define> weaselly </define> figure , a sallow complexion , a certain
obliquity of vision , and a large pair of spectacles .
Target text: Resembling a weasel (in appearance).
Generated text: Resembling or characteristic of a weasel.
Answer:
```

Figure 2: Example prompt for a Stage 2 classifier, given a Definition Modeling task data point from one of the SHROOM datasets, and using 1 example per label.

ing the LLM multiple times to generate a sample of classifications, and then dividing the number of positive classifications (i.e., where the generated label is ’Hallucination’) by the total number of classifications in the sample. Temperature sampling is performed in producing both Stage 1 zero-shot and Stage 2 few-shot classifications.

3.3 Example selection

In Stage 1, the algorithm processes an unlabeled dataset to generate examples using a zero-shot query. Following the Self-Adaptive Prompting approach described in (Wan et al., 2023a,b), for each task type we sample 64 data points from the unlabelled dataset, and then use a zero-shot query to obtain a classification with estimated probability of hallucination. This information is combined with the data point to produce an example. We partition the examples per task type into two pools, one with positive examples where the label is ’Hallucination’ and the other with negative examples where the label is ’Not Hallucination’.

The process used to select the examples to include in the prompt is shown in Algorithm 1. The first example chosen from each pool is the one with the maximum negative entropy of the classification probability, as defined in Equation 1:

$$F_0(p) = p * \log p + (1-p) * \log(1-p) \tag{1}$$

Algorithm 1 Select examples given a task and label

```
Require: $P$: generated examples for given task and label, $K$: number of selections
Ensure: $S$: selected examples
$S \leftarrow \emptyset$
$Pool \leftarrow P$
for $k \leftarrow 0$ to $K-1$ do
    if $k == 0$ then
        $s_k \leftarrow \arg\max_{p \in Pool} F_0(p)$
    else
        $s_k \leftarrow \arg\max_{p \in Pool} F(p, S)$
    end if
    $S \leftarrow S \cup \{s_k\}$
    $Pool \leftarrow Pool \setminus \{s_k\}$
end for
```

For each remaining selection $i \leq K$, the algorithm selects the example that maximizes a trade-off between the diversity of prompts and the consistency of the majority voting result, as defined in Equation 2:

$$F(p, S) = F_0(p) - \lambda \cdot \max_{s \in S} (1 - \text{sim}(\phi(p), \phi(s))) \tag{2}$$

ϕ is calculated for a given data point by concatenating its data into a string and then using an embedding model to produce a representation vector. This trade-off is quantified by subtracting a weighted maximum cosine similarity of the embeddings from the negative entropy, with the weight λ controlling the balance between diversity and consistency. In all of our experiments, in keeping with

(Wan et al., 2023b), λ is set to 0.2. The selected examples for both labels are then serialized and concatenated. This concatenated string is then used to augment the zero-shot query prompt given the task.

4 Experimental Setup and Results

The LLMs used in the evaluating the system were from OpenAI (`gpt-3.5-turbo`, `gpt-4-0125-preview`) and were invoked using the OpenAI API with the LangChain Python library. Stage 1 was performed once with $K = 5$ using `gpt-4-0125-preview` on 25 January 2024. The embedding model used in the calculation of ϕ was OpenAI `text-embedding-ada-002`. The Stage 2 run for our final submission during the evaluation period was conducted on 28 January 2024. Runs for the hyperparameter and ablation study results reported below were conducted between 17 February 2024 and 18 February 2024. Approximately $500 USD in OpenAI API charges were incurred during the above runs.

4.1 Classification performance

As shown in Table 3, using `gpt-4-0125-preview` and `gpt-3.5-turbo` as LLMs our approach showed a significant improvement in both accuracy and Spearman's ρ over the baseline reported for the model-agnostic and model-aware validation sets.[2]

Our best-performing submission to the competition used `gpt-4-0125-preview` as its LLM with 1 example provided per label, 20 samples for majority voting, and a temperature setting of 1.2. We compare it to the baseline system's performance on the test datasets together with that reported for each of the first ranked teams in the model-agnostic track (GroupCheckGPT) and the model-aware track (HaRMoNEE). The SHROOM-INDElab system ranked fourth and sixth in the tracks, respectively.

The values of ρ can be interpreted as showing a moderate to strong correlation between the estimated probability of hallucination provided by the system and that provided by the majority vote result of the human labellers.

4.2 Hyperparameter study

The classifier has three hyperparameters; temperature, which is the parameter passed to the language model to indicate the level of stochasticity associated with its generation process; the number of examples per label provided for in-context learning; and the number of samples per query performed and used to calculate the estimated probability associated with the classification of the data point.

We investigated the impact of varying the values of the three hyperparameters of the classifier on the classifier's performance. We used `gpt-3.5-turbo` to conduct this investigation, computing values of accuracy and Spearman's ρ by executing three different passes over the model-agnostic validation dataset.

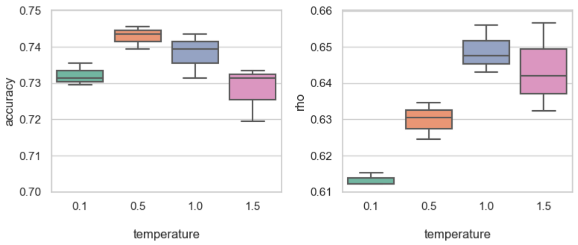


Figure 3: Classifier performance by temperature.

Figure 3 shows the best classifier accuracy is obtained with a temperature between 0.5 and 1.0, and that the best value for Spearman's ρ is obtained with a temperature between 0.5 and 1.5, given settings of 1 example per label and 5 samples per query.

Figure 4 shows that increasing the number of examples for few-shot classification beyond one per label led to an increase in accuracy with diminishing returns after 2 examples per label, but a decrease in Spearman's ρ, given settings of temperature of 1.0 and 5 samples per query.

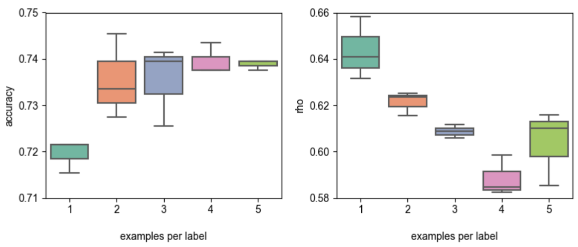


Figure 4: Classifier performance by examples per label.

Figure 5 shows that increasing the number of samples per query led to an increase in both accuracy and Spearman's ρ, given 1 example per label

[2] Although we submitted results for the model-aware track, our implementation of the approach is model agnostic and does not utilize the model field of the data point.

		model-agnostic		model-aware	
Dataset	**System**	accuracy	ρ	accuracy	ρ
Validation	Baseline	0.649 (+0.000)	0.380 (+0.000)	0.707 (+0.000)	0.461 (+0.000)
	SHROOM-INDElab (`gpt-3.5-turbo`)	0.773 (+0.124)	0.652 (+0.272)	0.764 (+0.057)	0.605 (+0.144)
	SHROOM-INDElab (`gpt-4-0125-preview`)	**0.814** (+0.165)	**0.697** (+0.317)	**0.772** (+0.065)	**0.635** (+0.174)
Test	Baseline	0.697 (+0.000)	0.403 (+0.000)	0.745 (+0.000)	0.488 (+0.000)
	SHROOM-INDElab (`gpt-4-0125-preview`)	0.829 (+0.132)	0.652 (+0.249)	0.802 (+0.057)	0.605 (+0.117)
	HaRMoNEE	0.814 (+0.117)	0.626 (+0.223)	**0.813** (+0.068)	0.699 (+0.210)
	GroupCheckGPT	**0.847** (+0.150)	**0.769** (+0.366)	0.806 (+0.061)	**0.715** (+0.227)

Table 3: Classifier performance on SHROOM datasets. ρ = Spearman's correlation coefficient.

and a temperature of 1.0.

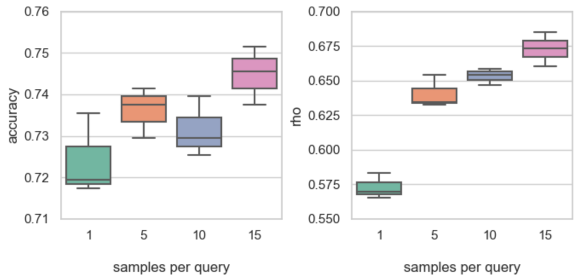


Figure 5: Classifier performance by samples per query.

4.3 Ablation study

Figure 6 shows the results of an ablation study to determine the contribution of the various elements of the prompt provided to the language models. We evaluated the contribution of each of the components of the Stage 2 classifier prompt by removing each in sequence, in the following order: the selected examples, the task definition, the role definition, and finally the concept definition. The ablation study was conducted using `gpt-3.5-turbo`, with 1 example per label, 5 samples per query, and a temperature of 1.0, again involving three different passes over the model-agnostic validation dataset.

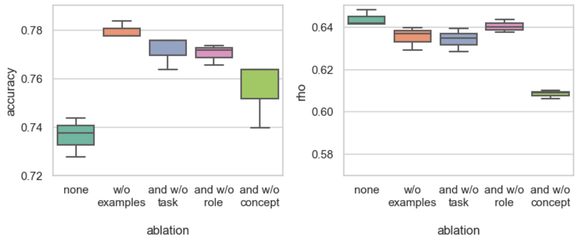


Figure 6: Ablation study using the model-agnostic validation dataset.

We interpret the results of the ablation study as indicating that the use of examples led to poorer accuracy but slightly better Spearman's ρ, that the contributions of the definitions of task and role towards classifier performance were minimal, but that the contribution of the definition of the concept of hallucination was significant.

4.4 Level of agreement with human labellers

We also investigated the degree of inter-annotator alignment exhibited with respect to the model-agnostic data set. Based on the human labeling data associated with each data point in the model-agnostic validation data set, we obtained a Fleiss' κ of 0.373, which can be interpreted as indicating a fair level of agreement among the human labellers, which in turn implies that the reliability of the human labeling might be reasonable, but is not highly consistent or unanimous. Adding the classifier's labeling yields an increase in Fleiss' κ to 0.405, closer to a moderate level of agreement, which implies that the classifier's decisions are consistent with those of the human labellers.

human consensus	**N**	**accuracy**	κ	ρ
low (2/3 split)	145	0.621	0.238	0.224
high (4/5 split)	171	0.854	0.701	0.734
unanimous	183	**0.929**	**0.856**	**0.885**
all	499	0.814	0.623	0.697

Table 4: Alignment between the system and human labellers.

We then proceeded to investigate the relationship between the degree of agreement between human labellers and system performance. Table 4 shows the level of agreement between the system and the human labellers, as measured by taking subsets of data points from the model-agnostic validation dataset filtered by the three degrees of consistency in human labeling and calculating the pairwise Cohen's κ between the system's labeling and the label provided by taking the majority vote of the human labellers. The results indicate that system agreement with human labeling increases as the certainty of the human labeling increases.

5 Discussion and Conclusion

In summary, the SHROOM-INDElab system was competitive with the other systems submitted for evaluation, and system labeling was consistent with that of human labellers.

The result in the ablation study that the exclusion of selected examples led to better accuracy suggests the need for further investigation with respect to how the way in which examples are selected and included in the classifier prompts impacts accuracy to determine the cause of the problem. The result that the exclusion of an explicit definition of hallucination leads to poorer accuracy and Spearman's ρ suggests the utility of including intentional definitions of concepts in prompts for LLM-based classifiers (Allen, 2023).

Given the above results, we plan to investigate the use of this approach to hallucination detection in future work on the evaluation of natural language rationale generation (Li et al., 2024) in the context of zero- and few-shot chain-of-thought classifiers for use in knowledge graph evaluation and refinement (Allen et al., 2023).

Acknowledgements

This work is partially supported by the European Union's Horizon Europe research and innovation programme within the ENEXA project (grant Agreement no. 101070305).

References

David H Ackley, Geoffrey E Hinton, and Terrence J Sejnowski. 1985. A learning algorithm for boltzmann machines. *Cognitive Science*, 9(1):147–169.

Bradley P Allen. 2023. Conceptual engineering using large language models. *arXiv preprint arXiv:2312.03749*.

Bradley P. Allen, Lise Stork, and Paul Groth. 2023. Knowledge Engineering Using Large Language Models. *Transactions on Graph Data and Knowledge*, 1(1):3:1–3:19.

Robert Friel and Atindriyo Sanyal. 2023. Chainpoll: A high efficacy method for llm hallucination detection. *arXiv preprint arXiv:2310.18344*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in Neural Information Processing Systems*, 35:22199–22213.

Aobo Kong, Shiwan Zhao, Hao Chen, Qicheng Li, Yong Qin, Ruiqi Sun, and Xin Zhou. 2023. Better zero-shot reasoning with role-play prompting. *arXiv preprint arXiv:2308.07702*.

Zhen Li, Xiaohan Xu, Tao Shen, Can Xu, Jia-Chen Gu, and Chongyang Tao. 2024. Leveraging large language models for nlg evaluation: A survey. *arXiv preprint arXiv:2401.07103*.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9).

Potsawee Manakul, Adian Liusie, and Mark JF Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *arXiv preprint arXiv:2303.08896*.

Timothee Mickus, Elaine Zosa, Raúl Vázquez, Teemu Vahtola, Jörg Tiedemann, Vincent Segonne, Alessandro Raganato, and Marianna Apidianaki. 2024. SemEval-2024 Task 6: SHROOM, a shared-task on hallucinations and related observable overgeneration mistakes. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, Mexico City, Mexico. Association for Computational Linguistics.

Fina Polat, Ilaria Tiddi, and Paul Groth. 2024. Testing prompt engineering methods for knowledge extraction from text. *Semantic Web*. Under Review.

Murray Shanahan, Kyle McDonell, and Laria Reynolds. 2023. Role play with large language models. *Nature*, pages 1–6.

Xingchen Wan, Ruoxi Sun, Hanjun Dai, Sercan O Arik, and Tomas Pfister. 2023a. Better zero-shot reasoning with self-adaptive prompting. *arXiv preprint arXiv:2305.14106*.

Xingchen Wan, Ruoxi Sun, Hootan Nakhost, Hanjun Dai, Julian Martin Eisenschlos, Sercan O Arik, and Tomas Pfister. 2023b. Universal self-adaptive prompting. *arXiv preprint arXiv:2305.14926*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.