uir-cis at SemEval-2025 Task 3: Detection of Hallucinations in Generated Text

Jia Huang[1], Shuli Zhao[2], Yaru Zhao[1], Tao Chen[1], Weijia Zhao[1], Hangui Lin[1], Yiyang Chen[1], Binyang Li [1]*

[1]University of International Relations, [2]Shanghai Jiao Tong University
{jiahuang, zhaoyaru, taochen, wjzhao, linhangui, uiryangyc0114, byli}@uir.edu.cn, shuli.zhao@sjtu.edu.cn

Abstract

The widespread deployment of large language models (LLMs) across diverse domains has underscored the critical need to ensure the credibility and accuracy of their generated content, particularly in the presence of hallucinations. These hallucinations can severely compromise both the practical performance of models and the security of their applications. In response to this issue, SemEval-2025 Task 3 Mu-SHROOM: Multilingual Shared-task on Hallucinations and Related Observable Overgeneration Mistakes introduces a more granular task for hallucination detection. This task seeks to identify hallucinations in text, accurately locate hallucinated segments, and assess their credibility. In this paper, we present a three-stage method for fine-grained hallucination detection and localization. First, we transform the text into a triplet representation, facilitating more precise hallucination analysis. Next, we leverage a large language model to generate fact-reference texts that correspond to the triplets. Finally, we employ a fact alignment strategy to identify and localize hallucinated segments by evaluating the semantic consistency between the extracted triplets and the generated reference texts. We evaluate our method on the unlabelled test set across all languages in Task 3, demonstrating strong detection performance and validating its effectiveness in multilingual contexts.

1 Introduction

LLMs have gained widespread application across various fields due to their exceptional performance, making them a core technology of significant interest. However, their reliance on vast training data and probabilistic inference mechanisms makes them prone to generating factually incorrect, misleading, or unsubstantiated content, leading to the phenomenon of "hallucination" (Bai et al., 2024). Hallucinations not only reduce the reliability and practicality of model-generated content but also pose safety and trust issues in real-world applications, ultimately affecting their effectiveness in critical domains. Therefore, accurately detecting and locating hallucinated information has become a key research challenge.

Existing research on hallucination detection has explored different granularity levels, including response-level detection (Zhou et al., 2020), which determines whether an entire output contains hallucinations; sentence-level detection (Mishra et al., 2024), which analyzes whether an individual sentence includes false information; and phrase-level detection (Min et al., 2023), which identifies hallucinations at a finer semantic unit. Although these methods have made progress in hallucination identification, most still struggle to accurately predict the exact location of hallucinations, particularly in multilingual settings. To bridge this gap, SemEval-2025 Task 3 Mu-SHROOM (Vázquez et al., 2025), introduces a more fine-grained hallucination detection and localization task, aiming to advance research on hallucination detection and localization across 14 languages.

For this task, we proposes a **three-stage method** for hallucination detection and localization in a passage. First, the method converts passages into multiple triple representations to enable more fine-grained hallucination detection. Next, for each extracted triple, the method generates fact-based reference text. Finally, by comparing the semantic similarity between the original triples and the generated fact-based references, the method identifies hallucinated triples, precisely locates the hallucinated segments, and formats the output according to the task requirements. The proposed method integrates fine-grained semantic analysis and fact-alignment strategies, allowing not only the identification of hallucinated information in passages but also the precise localization of hallucinations. In

*Corresponding author

Proceedings of the The 19th International Workshop on Semantic Evaluation (SemEval-2025), pages 1015–1022

the SemEval-2025 Task 3 competition, the method was tested on the unlabeled test sets for all languages and achieved corresponding detection results.

2 Related Work

2.1 Response-Level Detection

Existing response-level hallucination detection methods evaluate generated text as a whole to assess its factual consistency but face limitations in fine-grained hallucination identification and localization. For instance, TruthfulQA (Lin et al., 2021) evaluates the overall truthfulness of model-generated text and finds that larger models are more prone to generating misleading hallucinations, highlighting that increasing model size alone does not enhance truthfulness. HaluEval (Li et al., 2023) employs fine-grained annotations and a "sampling-filtering" mechanism to construct high-quality hallucination datasets, revealing distinct hallucination patterns of LLMs across different tasks. However, a major drawback of these methods lies in their inability to precisely locate hallucinated content. In SemEval-2025 Task 3 Mu-SHROOM, hallucination detection requires not only identifying hallucinations but also predicting their exact locations. Response-level detection methods, however, can only determine the overall factual consistency of a text without providing character-level annotations. Moreover, existing approaches struggle with generalization in multilingual and multimodal settings, making cross-lingual hallucination detection particularly challenging.

2.2 Sentence-Level Detection

Unlike response-level detection, sentence-level hallucination detection evaluates the truthfulness of generated text on a per-sentence basis to enable more fine-grained hallucination identification and enhance local factual consistency. For example, (Manakul et al., 2023) identify potential hallucinations by comparing multiple sampled responses to the same query and measuring their consistency. (Deng et al., 2024) propose PFME (Progressive Fine-Grained Model Editor), which integrates real-time fact retrieval with fine-grained editing to detect and correct hallucinations at the sentence level, improving both truthfulness and reliability. However, despite offering finer-grained analysis than response-level methods, sentence-level approaches remain limited by their reliance on evaluating entire sentences rather than precisely locating hallucinated segments. These methods often depend on inter-sentence consistency or external fact alignment but fail to pinpoint the exact position of hallucinations within a sentence.

2.3 Phrase-Level Detection

Advancing beyond sentence-level methods, phrase-level hallucination detection decomposes sentences into clause-level factual assertions, enabling more precise hallucination identification and correction while improving the detection of multiple hallucinations within a single sentence. For instance, FActScore (Min et al., 2023) evaluates the factual consistency of long-form text generation by segmenting generated text into a series of atomic facts and extracting phrases as claim units. It then calculates the proportion of claims supported by reliable knowledge sources to enhance evaluation accuracy. Additionally, (Chern et al., 2023) integrate tool-augmented methods with a fine-grained claim extraction mechanism, partitioning generated text into atomic content units (ACUs) to detect factual errors with greater precision. Although phrase-level methods offer finer granularity than sentence-level approaches, their primary limitation lies in weak localization capabilities. Most methods rely on claim unit segmentation and fact comparison but struggle to precisely align hallucinated content with specific entities or relations in the text.

Compared to existing response-level, sentence-level, and phrase-level hallucination detection methods, the proposed three-stage fine-grained detection and localization approach offers significant advantages in detection granularity and localization accuracy. Response-level methods assess the overall factual consistency of text but fail to precisely locate hallucinations, making them inadequate for SemEval-2025 Task 3 Mu-SHROOM, which requires character-level annotation. Sentence-level methods evaluate hallucinations on a per-sentence basis but still analyze entire sentences, making it difficult to identify the exact part of the sentence where hallucinations occur. Phrase-level methods, such as FActScore and ACUs, decompose sentences into claim units but rely on clause segmentation and fact comparison, which limits their ability to accurately align hallucinated content with specific entities or relations in the text.

The proposed method extracts triples to parse text into more granular subject-verb-object structures, integrating fact comparison and semantic

consistency computation to enable hallucination detection at the clause, phrase, and even word level. This approach ensures precise hallucination localization and outputs results in a structured format. Compared to existing methods, it demonstrates superior detection accuracy, localization capability, multilingual adaptability, and interpretability, making it better suited to meet the fine-grained hallucination detection and localization requirements of SemEval-2025 Task 3.

3 Task Description and Datasets

In Mu-SHROOM Task 3, the organizers introduce a task aimed at predicting the locations of hallucinated segments in text generated by LLMs. This task focuses on identifying hallucinations in LLM outputs across multiple languages, including Modern Standard Arabic, Basque, Catalan, Mandarin Chinese, Czech, English, Farsi, Finnish, French, German, Hindi, Italian, Spanish, and Swedish.

The dataset consists of a sample set, a validation set, and an unlabeled training set. The sample set includes model-generated question-answer pairs, soft labels, and hard labels. Soft labels indicate potential hallucinated spans along with their probability distributions, while hard labels specify the exact boundaries of hallucinated segments within the generated text. Compared to the sample set, the validation set provides additional tokenized outputs and logit values for each generated token. The unlabeled training set contains only model-generated question-answer pairs without hallucination annotations.

4 Methodology

We proposed a **three-stage method**, as illustrated in Figure 1. In the first stage, we applies the technique from RefChecker (Hu et al., 2024) to extract multiple triples from each user-provided passage. In the second stage, a high-performing language model generates references for each extracted triple based on stored knowledge. In the third stage, the method compares the generated references with the original triples to assess their semantic similarity, determine whether hallucinations are present, and perform hallucination localization and structured output formatting.

4.1 Triplet Generation

A triple serves as a structured representation of knowledge, fundamentally consisting of an ordered

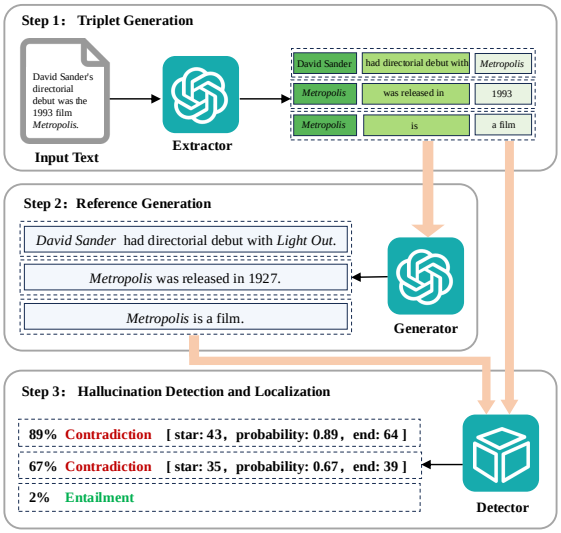

Figure 1: The overview of the three-stage hallucination detection process.

list with three elements: the subject, predicate, and object extracted from a sentence. The subject represents an entity and serves as the core component of a sentence, indicating the entity being described. The predicate expresses the relationship or attribute between the subject and object, typically consisting of a verb or verb phrase that conveys the subject's actions, state, or characteristics. The object functions as the target of the predicate and can represent an entity. Together, the subject, predicate, and object form a complete semantic expression.

A triple extracts the subject, predicate, and object from a sentence and structures them into a formal representation, enabling more effective identification of core information within the text. The triple generation module processes each input passage under evaluation and extracts one or more triples from it.Specifically, the triple generation module employs high-capability language models, such as GPT-4 (Achiam et al., 2023) and DeepSeek (Guo et al., 2025), to process passages in batches and convert them into one or more triples. The extracted triples encompass all key factual relationships present in the passage. The prompt used is included in the Appendix.

4.2 Reference Generation

The reference generation module constructs references based on the content of the passage under evaluation. Specifically, the reference generation module constructs prompts based on the subject-predicate-object triples extracted by the triple generation module and utilizes an advanced language

model with internet access, such as GPT-4 (Achiam et al., 2023) or DeepSeek (Guo et al., 2025), to generate the corresponding references. The prompt used is included in the Appendix. In our design, the language model evaluates the accuracy of the triples by leveraging both internet-accessed information and its internal knowledge, generating corresponding reference text accordingly. To ensure the accuracy of hallucination detection, the generated reference text must precisely and correctly describe the content of the triples under evaluation.

Therefore, we impose constraints on the format of the generated reference text, allowing only two predefined structures:(1)If the content described by the triple aligns with factual information, the module directly converts the triple into a grammatically correct declarative sentence as the reference text. (2)If the triple contains inaccuracies or contradictions, the module retrieves factual information through the language model's internet access and reconstructs a factually accurate declarative sentence using the subject, predicate, and the correct object from the triple. Experimental results indicate that constraining the reference text format improves hallucination detection performance compared to an unconstrained approach.

4.3 Hallucination Detection and Localization

The hallucination detection and localization module utilizes the FacebookAI/roberta-large language model (Trinh and Le, 2018) to assess the semantic similarity between triples and the reference. Specifically, the model processes the input text in batches and applies softmax normalization to compute the probability distribution of relation matching, quantifying the semantic consistency between triples and the reference text. The essence of this task lies in multi-class classification, where the results of semantic similarity comparison are categorized into Entailment, Neutral, and Contradiction. If the text under evaluation contains hallucinations, a semantic discrepancy should exist between the triple and the corresponding reference, leading to a higher probability of being classified as Contradiction. Subsequently, the probability of being classified as Contradiction is used as the soft label, and samples with a predicted Contradiction probability greater than 0.5 are identified as containing hallucinations. Once hallucinated samples are determined, hallucination localization is conducted. Hallucination localization process focuses on the triples marked as hallucinations, identifying the position of their objects within the original passage and converting the results into a structured output format. In a sentence, the subject is typically a stable entity, the predicate expresses the relationship between the subject and the object, and the object is an entity or concept determined under the constraints of the given subject and predicate. In natural language, once the subject and predicate are established, the selection of the object usually belongs to an open set. When predicting the object entity, overgeneralization or erroneous associations may occur, leading to hallucinations. To facilitate hallucination localization, this module primarily focuses on identifying the position of hallucinated objects within a sentence while preventing errors caused by multiple occurrences of the same object. Finally, after determining the precise location for the hard label output, the module integrates the soft label value to generate a structured output, producing the final task-formatted result.

5 Experiments & Results

5.1 Implementation

Our method primarily employs API calls to interact with the large model and obtain its feedback. Specifically, during both the triple generation and reference generation steps, the GPT-4o model is accessed via API calls to generate triples and corresponding references. The hallucination detection module utilizes a locally deployed FacebookAI/roberta-large language model. To ensure efficient execution of experiments, all computations run in a GPU-accelerated environment. Specifically, NVIDIA RTX 4090 GPUs provides computational acceleration, while FP16 mixed-precision computation optimizes processing efficiency. All experiments run in the Ubuntu 22.04 operating system environment, with Python dependencies including Transformers[1], PyTorch[2], and related deep learning frameworks. To ensure stable API calls, the batch processing approach manages request execution, while appropriate rate limits prevent exceeding the API threshold and maintain experimental integrity.

5.2 Metrics

Our experiment employs two metrics, **Intersection-over-Union (IoU)** and **Spearman Correlation**

[1] https://github.com/huggingface/transformers
[2] https://github.com/pytorch/pytorch

(Cor), to assess the effectiveness of our method in hallucination detection.

IoU measures the ratio of the intersection to the union between the predicted hallucinated content and the ground truth hallucinated content, providing a quantitative evaluation of the overlap between the predicted and actual hallucination regions. Equation 1 presents the formula for IoU, where P denotes the hallucinated content predicted by the model, and G represents the ground truth hallucinated content identified through manual annotation.

$$\text{IoU} = \frac{|P \cap G|}{|P \cup G|} \tag{1}$$

Spearman correlation quantifies the rank correlation between the predicted probability scores of hallucination and the manually assigned scores, serving as a measure of consistency between the model's hallucination predictions and human evaluations. Equation 2 defines the Spearman correlation, where $\mathbf{p} = [p_1, p_2, \ldots, p_n]$ represents the sequence of hallucination probabilities predicted by the model, with p_i denoting the probability of hallucination at the i-th character. Similarly, $\mathbf{g} = [g_1, g_2, \ldots, g_n]$ denotes the sequence of hallucination proportions from human annotations, where g_i indicates the probability that the i-th character is annotated as hallucinated. The term d_i represents the rank difference between the predicted probability p_i and the human-annotated probability g_i after sorting. The variable n denotes the length of the text.

$$\rho = 1 - \frac{6 \sum_{i=1}^{n} d_i^2}{n(n^2 - 1)} \tag{2}$$

5.3 Results and Analysis

Our method is evaluated on the unlabeled test sets provided by the organizers across all languages, achieving a certain level of detection performance. Table 1 below presents the results of our approach on test sets for each language.

Our method offers advantages by incorporating triple generation, factual alignment, and semantic similarity computation, enabling precise detection and localization of hallucinated segments. Compared to traditional binary classification approaches and certain hallucination detection methods based on entire passages, our method enhances interpretability and achieves more fine-grained hallucination detection.

Language	IoU	Cor
AR	0.2722	0.4477
CA	0.4644	0.5432
CS	0.3060	0.2695
DE	0.3400	0.4066
EN	0.4025	0.4781
ES	0.3447	0.3104
EU	0.2916	0.3989
FA	0.1661	0.3946
FI	0.2459	0.3366
FR	0.2286	0.2873
HI	0.0613	0.5586
IT	0.3967	0.4991
SV	0.3080	0.3655
ZH	0.1913	0.3047

Table 1: Performance of our proposed method on different language test sets.

The experimental results reveal significant performance differences across languages in terms of IoU and Cor metrics. Specifically, the model achieves the best performance on the Catalan test set, with an IoU of 0.4644 and a Cor of 0.5432. On test sets for languages such as English and Italian, the IoU and Cor values remain stable at approximately 0.4 and 0.5, respectively, indicating strong generalization capability across these languages. However, on test sets for languages such as Farsi and Chinese, the IoU value falls below 0.2, which may be attributed to weaker generalization or higher linguistic complexity. Therefore, it may be worth considering the use of XLM-RoBERTa model specifically for these languages in the future.

6 Conclusion

This study proposes a three-stage approach for hallucination detection and localization, consisting of triple generation, reference text generation, and hallucination detection and localization. The method enables fine-grained identification of hallucinated content in text generated by large language models. Experimental results indicate that the method achieves good hallucination detection performance in languages such as English, but its detection accuracy decreases in some low-resource languages and those with complex syntactic structures and data.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Zechen Bai, Pichao Wang, Tianjun Xiao, Tong He, Zongbo Han, Zheng Zhang, and Mike Zheng Shou. 2024. Hallucination of multimodal large language models: A survey. *arXiv preprint arXiv:2404.18930*.

I Chern, Steffi Chern, Shiqi Chen, Weizhe Yuan, Kehua Feng, Chunting Zhou, Junxian He, Graham Neubig, Pengfei Liu, et al. 2023. Factool: Factuality detection in generative ai–a tool augmented framework for multi-task and multi-domain scenarios. *arXiv preprint arXiv:2307.13528*.

Kunquan Deng, Zeyu Huang, Chen Li, Chenghua Lin, Min Gao, and Wenge Rong. 2024. Pfme: A modular approach for fine-grained hallucination detection and editing of large language models. *arXiv preprint arXiv:2407.00488*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Xiangkun Hu, Dongyu Ru, Lin Qiu, Qipeng Guo, Tianhang Zhang, Yang Xu, Yun Luo, Pengfei Liu, Yue Zhang, and Zheng Zhang. 2024. Knowledge-centric hallucination detection. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6953–6975.

Junyi Li, Xiaoxue Cheng, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023. Halueval: A large-scale hallucination evaluation benchmark for large language models. *arXiv preprint arXiv:2305.11747*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2021. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*.

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*.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. *arXiv preprint arXiv:2305.14251*.

Abhika Mishra, Akari Asai, Vidhisha Balachandran, Yizhong Wang, Graham Neubig, Yulia Tsvetkov, and Hannaneh Hajishirzi. 2024. Fine-grained hallucination detection and editing for language models. *arXiv preprint arXiv:2401.06855*.

Trieu H Trinh and Quoc V Le. 2018. A simple method for commonsense reasoning. *arXiv preprint arXiv:1806.02847*.

Raúl Vázquez, Timothee Mickus, Elaine Zosa, Teemu Vahtola, Jörg Tiedemann, Aman Sinha, Vincent Segonne, Fernando Sánchez-Vega, Alessandro Raganato, Jindřich Libovický, Jussi Karlgren, Shaoxiong Ji, Jindřich Helcl, Liane Guillou, Ona de Gibert, Jaione Bengoetxea, Joseph Attieh, and Marianna Apidianaki. 2025. SemEval-2025 Task 3: Mu-SHROOM, the multilingual shared-task on hallucinations and related observable overgeneration mistakes.

Chunting Zhou, Graham Neubig, Jiatao Gu, Mona Diab, Paco Guzman, Luke Zettlemoyer, and Marjan Ghazvininejad. 2020. Detecting hallucinated content in conditional neural sequence generation. *arXiv preprint arXiv:2011.02593*.

A Prompts Used in the Method

The following presents the prompts used in this study for the triple generation and reference generation.

Prompt

Given an input text, please extract a KG from the text and represent the KG with triples formatted with ("subject", "predicate", "object"), each triplet in a line. Please note that this is an EXTRACTION task, so DO NOT care about whether the content of the candidate answer is factual or not, just extract the triplets from it. Importantly, ensure that the extracted KG does not contain overlapping or redundant information. Each piece of information should be represented in the KG only once, and you should avoid creating triplets that are simply the inverse of another triplet. For example, if you extract the triplet ("John", "owns", "Car"), do not also include ("Car", "owned by", "John") as it represents the same information in reverse. The language you generated should be the same as the language of the user input text.

Clarification on redundancy: First, Do not create triplets that reverse the subject and object to state the same fact. Next, Ensure each fact is represented uniquely in the simplest form, and avoid creating multiple triplets that convey the same information.

Here are some in-context examples:

Input:

Optimus (or Tesla Bot) is a robotic humanoid under development by Tesla, Inc. It was announced at the company's Artificial Intelligence (AI) Day event on August 19, 2021.

KG:

("Optimus", "is", "robotic humanoid")

("Optimus", "under development by", "Tesla, Inc.")

("Optimus", "also known as", "Tesla Bot")

("Tesla, Inc.", "announced", "Optimus")

("Announcement of Optimus", "occurred at", "Artificial Intelligence (AI) Day event")

("Artificial Intelligence (AI) Day event", "held on", "August 19, 2021")

("Artificial Intelligence (AI) Day event", "organized by", "Tesla, Inc.")

Input:

The song "Here Comes the Boom" was originally released by American rock band Nelly in 2002 for the soundtrack of the film "The Longest Yard."

KG:

("The song 'Here Comes the Boom'", "originally released by", "American rock band Nelly")

("The song 'Here Comes the Boom'", "released in", "2002")

("The song 'Here Comes the Boom'", "featured in", "soundtrack of the film 'The Longest Yard'")

("American rock band Nelly", "released", "The song 'Here Comes the Boom'")

("The Longest Yard", "had soundtrack featuring", "The song 'Here Comes the Boom'")

Now generate the KG for the provided input text:

Input:

{input_text}

KG:

Figure 2: Prompt of triplet generation.

Prompt

System Prompt : **You are a knowledgeable intelligent information retrieval assistant dedicated to providing users with accurate and valuable answers. For user inquiries, you will rely on your existing knowledge to respond and conduct searches when necessary to offer the most up-to-date and comprehensive information. Users may input statements that could be either correct or incorrect, and you need to provide supporting references or sources for the user's input without outputting any irrelevant content. Note that the language provided by the user may not be English, and you should respond in the language provided by the user.**

User Prompt :

Subject: {subject}, Predicate: {predicate}, Object : {object}. This statement may be correct or incorrect. If the statement is correct, repeat the statement in the format: subject + predicate + object. If the statement is incorrect, do not provide any reasons; simply output the correct statement by keeping the subject and predicate while replacing the object with the correct one. Do not provide any additional content, explanations, or reasons. You also do not need to indicate whether the statement is correct.

Example 1:

User input: Subject: A, Predicate: is, Object: B.

If the statement is correct, you should response: A is B.

If the statement is incorrect, you should response: A is C (where C is the correct answer for subject A and predicate "is").

Example 2:

User input: Subject: A, Predicate: contains, Object: B.

If the statement is correct, you should response: A contains B.

If the statement is incorrect, you should response: A contains C (where C is the correct answer for subject A and predicate "contains").'" + "Note that the language provided by the user may not be English, and you should respond in the language provided by the user.

Figure 3: Prompt of reference generation.