Enhancing Retrieval-Augmented Generation Systems with Hypothesis Testing

Dr. Gopichand Agnihotram[1], Neha Maurya[2], Joydeep Sarkar[3]
[1,2,3] *Wipro Limited*

***Abstract*—As large language models (LLMs) are increasingly embedded into knowledge-intensive applications through Retrieval-Augmented Generation (RAG), ensuring that both retrieval and response generation are meaningful and reliable becomes critical. Hypothesis testing, a foundational tool in inferential statistics, offers a principled framework to assess and control the reliability of retrieved contexts and generated outputs. This white paper explores how hypothesis testing can be integrated into RAG systems to improve contextual relevance, reduce hallucinations, and enhance the alignment between system outputs and source-grounded truth. We present a mathematically motivated foundation, practical formulations, and use-case driven illustrations for embedding statistical rigor into retrieval and generation.**

***Index Terms*—Retrieval-Augmented Generation (RAG), Hypothesis Testing, Large Language Models (LLMs), Hallucination Reduction, Natural Language Inference (NLI), Statistical Reliability**

I. INTRODUCTION

In the evolving landscape of artificial intelligence and natural language processing, large language models (LLMs) have demonstrated remarkable proficiency in generating fluent, coherent, and seemingly knowledgeable text. Despite their success, these models operate with a significant limitation: they are confined to the data seen during training and lack real-time access to external or dynamic sources of information. This constraint has led to the emergence of hybrid systems that combine language models with information retrieval mechanisms to bridge this gap [1]. Retrieval-Augmented Generation (RAG) is one such paradigm that aims to equip LLMs with the ability to fetch and utilize relevant external documents to enhance their outputs [2].

RAG systems function by first retrieving documents from a large corpus based on a user query, and then conditioning the generation of responses on this retrieved context. This approach holds significant promise, particularly in domains requiring up-to-date, domain-specific, or highly factual information, such as law, medicine, science, and finance. The integration of retrieval with generation introduces a layer of grounding that mitigates the tendency of LLMs to hallucinate facts or produce content that lacks verifiable sources [3].

However, while RAG systems mark a substantial improvement over standalone LLMs, they are not without challenges. The retrieval component, typically driven by embedding-based similarity search or sparse keyword matching, often returns documents that are only loosely related to the original query. These marginally relevant documents can dilute the context provided to the generator, leading to outputs that are less focused, misleading, or even incorrect. Additionally, the generator itself may misinterpret retrieved content or synthesize information in ways that are logically or factually invalid. The compounded effect of imperfect retrieval and imperfect generation undermines the reliability of RAG systems, particularly in use cases where precision and factual correctness are paramount.

To address these limitations, we propose the integration of hypothesis testing a cornerstone of statistical inference into the RAG architecture. Hypothesis testing provides a formal and interpretable method for making decisions under uncertainty. By treating the inclusion of a document or a claim as a testable hypothesis, systems can rely on probabilistic reasoning rather than heuristic scoring or fixed thresholds. For example, instead of simply ranking documents by cosine similarity and selecting the top-k, we can test whether the similarity score of a document is significantly different from that of irrelevant or noisy examples. Similarly, generated claims can be assessed based on whether they are statistically supported by the retrieved context.

This statistical framing transforms the RAG pipeline from a purely score-driven pipeline to a more principled, inference-based system. Each document and claim becomes subject to formal evaluation, increasing transparency and traceability in the decision-making process. Moreover, hypothesis testing facilitates dynamic adaptation-thresholds can be tuned according to domain requirements, risk tolerance, or user feedback, providing a flexible mechanism for balancing strictness with coverage.

Importantly, the adoption of hypothesis testing does not require discarding existing retrieval or generation techniques. Instead, it introduces a validation layer that augments and regulates their outputs. This layer can be lightweight and efficient, leveraging sampling, bootstrapping, or precomputed distributions to ensure scalability. When implemented effectively, it improves both the quality of retrieved evidence and the faithfulness of generated responses.

In summary, the motivation behind incorporating hypothesis testing into RAG systems lies in the pursuit of greater reliability, interpretability, and rigor in knowledge-intensive Al applications. As RAG continues to gain traction in industry and research, statistical tools like hypothesis testing will play a crucial role in bridging the gap between raw model performance and trustworthy, evidence-based Al outputs.

II. MATHEMATICAL FOUNDATION OF HYPOTHESIS TESTING

Hypothesis testing is a statistical method used to evaluate assumptions or claims about a population, typically framed in terms of two competing hypotheses: the null hypothesis and the alternative hypothesis [4]. The null hypothesis represents the status quo or a baseline assumption (e.g., "a document is not relevant to the query"), while the alternative represents a competing claim (e.g., "the document is relevant"). The core goal is to assess, using observed data, whether there is sufficient evidence to reject the null hypothesis in favour of the alternative.

The standard procedure of hypothesis testing involves the following key components:

Test Statistic (T): This is a numerical summary of the data used to determine how extreme the observed results are under the null hypothesis. In the RAG context, test statistics can be derived from similarity scores (between a query and retrieved documents) or entailment/confidence scores (between generated claims and supporting context).

Null Hypothesis (H_0): This defines the distribution of the test statistic under the assumption that the retrieved document or generated statement is not relevant or not supported. The distribution may be empirically constructed from irrelevant document-query pairs or unsupported claim-evidence combinations.

Alternative Hypothesis (H_1): This states that the test statistic comes from a distribution indicative of relevance or factual support. In practice, may be estimated from known relevant examples or annotated entailments.

Significance Level: This is the threshold for deciding whether an observed test statistic is extreme enough to reject the null hypothesis. A common value is $\alpha = 0.05$, representing a 5% chance of incorrectly rejecting the null hypothesis when it is true (Type I error).

Decision Rule: If the observed test statistic yields a p-value, the system rejects the null hypothesis and accepts that the document is relevant or that the generated claim is supported.

The p-value itself is defined as the probability of obtaining a test statistic as extreme or more extreme than the one observed, assuming the null hypothesis is true.

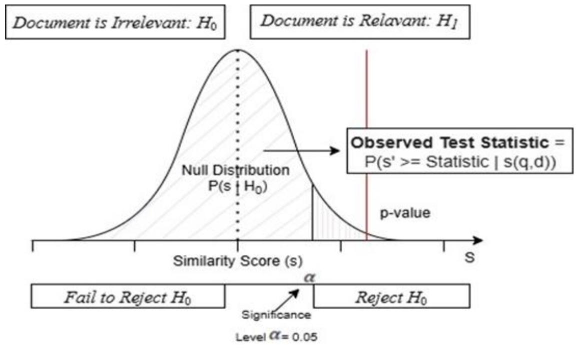
Document is Irrelevant: H_0
Document is Relavant: H_1
Observed Test Statistic = P(s' >= Statistic | s(q,d))
Null Distribution P(s | H_0)
p-value
Similarity Score (s)
S
α
Fail to Reject H_0
Reject H_0
Significance Level α= 0.05

Fig. 1. Illustration of the Hypothesis Testing Framework.

Fig. 1. explains hypothesis testing. The bell curve represents the Null Distribution (P(s | H_0) showing the expected spread of similarity scores if a document were irrelevant (H_0). The red vertical line marks the Observed Test Statistic (e.g., a document's similarity score, s(q, d)). The area to its right is the p-value, indicating the probability of observing such an extreme score if the document were truly irrelevant. If this p-value falls below the Significance Level (a, typically 0.05), we reject H_0, concluding the document is statistically relevant.

This provides a way to quantify how surprising the observed data is under the assumption that holds. A small p-value indicates strong evidence against the null.

In the context of RAG, these definitions can be mapped to operational tasks as follows:

- For document retrieval, could be the cosine similarity between the query vector and document embedding. The null hypothesis posits that this document is irrelevant, and the system computes the p-value by comparing to a null distribution of similarities from irrelevant pairs.
- For generation validation, may be the output of an NLI (Natural Language Inference) model, estimating the probability that a claim is entailed by retrieved evidence. A low entailment score corresponds to high p-values, suggesting lack of support.

Importantly, the test statistic must be calibrated and context-sensitive. For example, a similarity score of 0.6 might be meaningful in one domain but too low in another. Therefore, constructing robust reference distributions is essential. This can be accomplished through bootstrapping from background corpora, using known irrelevant documents, or leveraging domain-specific benchmarks.

This formal structure provides not just a decision rule, but a confidence metric—something that is critical in high-stakes applications. Moreover, because hypothesis testing can be repeated for every retrieved document or generated statement, the RAG system becomes equipped with a systematic way of adjudicating relevance and support across all stages of inference.

III. HYPOTHESIS TESTING IN DOCUMENT RETRIEVAL

The retrieval component of a RAG system typically ranks documents using similarity metrics derived from dense or sparse retrievers. These scores, such as cosine similarity are used to determine which documents to include in the top-k set that conditions the language model. Traditionally, these scores are treated deterministically: documents with the highest similarity scores are selected without further statistical reasoning. However, this approach lacks a formal mechanism for determining whether a document's similarity score reflects genuine relevance or is a result of random alignment in the embedding space.

To enhance rigor and reliability, we propose applying hypothesis testing to the document retrieval stage of the RAG pipeline. Let a query and a document be represented in an embedding space and let denote their cosine similarity. We pose the following statistical hypotheses:

- Null Hypothesis(H_0): Document is irrelevant to query.
- Alternative Hypothesis(H_1): Document is relevant to query.

The observed similarity score is evaluated against an empirical null distribution composed of similarity scores from known irrelevant documents:

This null distribution may be constructed from a background corpus using randomly sampled documents that are presumed irrelevant. In more refined setups, curated negative examples or domain-specific irrelevant corpora may be used.

Once is defined, the p-value associated with document d is calculated as:

$$p = P(s(q, d) \geq s_{obs} \mid H_0)$$

That is, the p-value reflects the probability of observing a similarity score as extreme as under the null hypothesis that the document is irrelevant. If the p-value is less than a predetermined significance level (e.g., 0.05), we reject the null hypothesis and consider the document relevant.

This method introduces statistical confidence into document selection. Rather than arbitrarily selecting the top-k documents, the system includes only those documents that pass a relevance test at a given confidence level. This can drastically reduce noise in the input to the generation model.

A central challenge in this approach is building an effective empirical distribution. Several strategies may be employed:

i. Random Sampling: Uniformly sample documents from a large corpus unrelated to the query. This provides a general baseline.
ii. Topic-Sensitive Sampling: Sample documents from categories or clusters known to differ from the query's intent.
iii. Bootstrapping: Use bootstrap resampling to estimate a distribution of similarity scores from presumed irrelevant documents.

The quality and representativeness of directly affect the reliability of the p-value calculations. In domains with abundant labelled data, supervised methods can be used to better identify irrelevant samples.

IV. BOOTSTRAPPING AND PERMUTATION TESTING FOR RELEVANCE ASSESSMENT

While hypothesis testing provides a principled framework for making inclusion decisions, the validity of these tests hinges on the availability of an appropriate null distribution. In many real-world RAG systems, this distribution is not known analytically. To address this, we can turn to non-parametric statistical methods, particularly bootstrapping and permutation testing, to empirically approximate and derive meaningful p-values [5].

A. Bootstrapping: Sampling the Empirical Null

Bootstrapping is a powerful resampling method used to approximate the distribution of a test statistic when its analytical form is unknown or difficult to derive [6], In the RAG setting, suppose we have a background set of similarity scores obtained from known or presumed irrelevant documents.

To assess the relevance of a new document with observed similarity score, we use the following procedure:

i. Resampling: Draw bootstrap samples, each of size, from the original similarity scores with replacement.
ii. Test Statistic Computation: For each bootstrap sample, compute the maximum (or mean) similarity score.
iii. Empirical P-Value Estimation: Estimate the p-value as the proportion of bootstrap test statistics at least as extreme as the observed similarity score i.e. (p = (number of bootstrap samples with test statistic ≥ observed score) / total number of bootstrap samples).

This process yields a robust, data-driven estimate of the probability that could arise under the null hypothesis. Since no parametric assumptions are made, bootstrapping is especially useful in high-dimensional embedding spaces where similarity scores may not follow familiar distributions.

B. Permutation Testing: Simulating Irrelevance

Permutation testing provides another non-parametric method for estimating null distributions, especially when bootstrapping from irrelevant examples is insufficient or unavailable. The idea is to generate synthetic irrelevant pairs by randomly permuting the association between queries and documents. These permuted pairs simulate what the distribution of similarity scores might look like under the null hypothesis.

Here's how permutation testing can be applied in the RAG retrieval context:

A. Generate Null Samples: Construct permuted pairs, where each is randomly selected or reshuffled from a set of documents unrelated to the original query.
B. Compute Test Statistics: For each permuted pair, compute the similarity score.
C. Estimate the P-Value: Compare the observed similarity to the permutation distribution.

This method constructs from simulated irrelevant data and enables hypothesis testing even in domains without labelled irrelevant documents.

V. HYPOTHESIS TESTING IN LLM RESPONSE VALIDATION

While retrieval quality in RAG systems determines the scope of available information, the final response generated by the language model must accurately reflect and remain grounded in this evidence. A major challenge in current RAG pipelines is the hallucination of facts or claims that are not supported-or only weakly implied by the retrieved documents [7], Hypothesis testing provides a formal statistical framework to evaluate whether a generated statement is sufficiently supported by its evidentiary context.

A. Formulating the Hypothesis

Each generated claim or sentence in a response can be cast as a hypothesis:

- Null Hypothesis: The claim is not supported by any of the retrieved documents.
- Alternative Hypothesis: The claim is entailed by or supported by at least one document.

We define a support function, where represents the set of retrieved documents, and returns a scalar entailment score. This score reflects how strongly the documents support the claim. The support function can be implemented using a textual entailment classifier, such as a BERT model fine-tuned on SNLI (Stanford Natural Language Inference) or MultiNLI datasets, which outputs a probability that the document entails the claim [8].

The test statistic for hypothesis testing becomes, $s(c, D) = \max_{d \in D} \text{entailment\_score}(c, d)$

where:

- D is the set of retrieved documents.
- s(c, D): The support score for claim c given the set of retrieved documents D.
- $\max_{d \in D}$: The maximum is taken over all documents d in the set D.
- $\text{entailment\_score}(c, d)$: A function that returns how strongly document d supports or entails claim c, typically derived from a Natural Language Inference (NLI) model.

This score may represent the maximum entailment score across all, or an aggregate like the weighted average or median, depending on the chosen strategy.

B. Estimating the Null Distribution

To compute a p-value, we need a null distribution representing entailment scores for unsupported claims. This can be constructed empirically by evaluating entailment scores between randomly paired claims and documents known to be irrelevant or mismatched. For instance, we can:

- Shuffle claims with unrelated documents.
- Sample sentences from the model output and match them with out-of-domain texts.
- Use known counterfactuals or adversarial examples with no valid support.

From these mismatched pairs, we compute entailment scores and derive the empirical distribution of those scores under the null hypothesis.

C. Hypothesis Testing Decision Rule

Once the observed test statistic is available, the p-value is computed. If the p-value is less than the significance level (typically $\alpha = 0.05$), we reject the null hypothesis and conclude that the claim is statistically supported by the retrieved evidence. Otherwise, the claim is deemed insufficiently grounded and may be:

- Flagged for review.
- Revised using contrastive decoding or paraphrasing.
- Removed entirely if unsupported.

D. Strategies for Multiple Claims

A typical RAG response may include several claims, each needing independent validation. Hypothesis testing can be applied at the sentence or clause level, with p-values computed for each unit. However, this introduces the risk of multiple testing errors (Type I errors).

To address this, multiple hypothesis correction techniques like the Bonferroni correction or False Discovery Rate (FDR) control (e.g., Benjamini-Hochberg procedure) can be employed to maintain global significance thresholds [9]:

- For total claims, Bonferroni-adjusted threshold becomes $\alpha' = \alpha / m$
- FDR control allows adaptive thresholds while controlling for the expected proportion of false discoveries.

VI. MULTIPLE HYPOTHESIS CORRECTION: ENSURING GLOBAL RELIABILITY

The framework detailed in Section 5 provides a robust method for validating a single generated claim against a set of retrieved documents. However, a typical RAG system response is rarely a single statement; it is often a paragraph composed of multiple distinct factual claims. Applying the hypothesis test independently to each of these claims introduces a significant statistical challenge known as the multiple testing problem. When multiple tests are run, the probability of making at least one Type 1 error (a false positive or incorrectly marking an unsupported claim as "supported") across the entire set of claims increases dramatically. Therefore, to maintain statistical integrity for the entire response, we must adjust our methodology using formal correction procedures.

Let me be the number of hypotheses (claims tested). Bonferroni sets a new threshold ($a' = a/m$) for each test. FDR methods allow more sensitivity while controlling the expected proportion of false discoveries.

VII. REAL-WORLD EXAMPLE: A STEP-BY-STEP APPLICATION IN THE LEGAL DOMAIN

Consider a legal assistant powered by a RAG system designed to help a paralegal with preliminary case research. The goal is to ensure the generated information is not only relevant but also statistically grounded in reliable legal sources.

Query: "What is the statute of limitations for a breach of contract claim under New York law?"

Step 1: Initial Retrieval

The system's retriever, based on a sentence-transformer model, searches a vector database containing New York State statutes, case law, and legal commentary. It identifies five potentially relevant document chunks based on high cosine similarity scores.

- Doc A: NY CPLR § 213 (Similarity: 0.92)
- Doc B: A specific case, ACE Sec. Corp. v. DB Structured Prods., Inc. (Similarity: 0.88)
- Doc C: A law blog post discussing contract law (Similarity: 0.81)
- Doc D: A case summary related to personal injury (Similarity: 0.75)
- Doc E: A statute regarding real estate transactions (Similarity: 0.72)

Step 2: Hypothesis Testing in Retrieval

Instead of just taking the top three documents, the system tests the relevance of each one. The null hypothesis (H_0) is "This document is irrelevant to the query." The system's empirical null distribution, built from thousands of random query-document pairs, indicates that for this legal domain, a similarity score of 0.75 has a p-value of 0.10, while a score of 0.80 has a p-value of 0.04.

- Doc A (Sim: 0.92): The p-value is calculated as $p < 0.001$ We reject H_0. Result: Relevant.
- Doc B (Sim: 0.88): The p-value is calculated as $p = 0.005$ We reject H_0. Result: Relevant.
- Doc C (Sim: 0.81): The p-value is calculated as $p = 0.03$ We reject H_0. Result: Relevant.
- Doc D (Sim: 0.75): The p-value is calculated as $p = 0.1$ We fail to reject H_0. Result: Not statistically relevant.
- Doc E (Sim: 0.72): The p-value is calculated as $p = 0.15$ We fail to reject H_0. Result: Not statistically relevant.

The context passed to the LLM now consists only of Docs A, B, and C, effectively filtering out noise from tangentially related documents like the one on personal injury.

Step 3: Response Generation

The LLM receives the curated context (Docs A, B, C) and the original query. It generates the following multi-claim response:

"Under New York law, the statute of limitations for a breach of contract is generally six years, as established in NY CPLR § 213. This period begins to run when the cause of action accrues. For contracts involving the sale of goods, the period is four years."

Step 4: Hypothesis Testing in Response Validation

The system now validates each factual claim in the response. The null hypothesis (Ha) for each claim is "This claim is not supported by the provided context." The test statistic is an entailment score from a fine-tuned NLI model.

- Claim 1: “the statute of limitations for a breach of contract is generally six years, as established in NY CPLR § 213.”
- Evidence: Doc A explicitly states this.
- Entailment Score: 0.97.
- P-value (from null distribution of entailment scores): $p < 0.001$, we reject H_0.

Result: Supported.

- Claim 2: “This period begins to run when the cause of action accrues.”
- Evidence: Doc B (the case law) discusses this principle.
- Entailment Score: 0.91.
- P-value: $p = 0.01$, we reject H_0.

Result: Supported

- Claim 3: “For contracts involving the sale of goods, the period is four years.”
- Evidence: None of the provided documents (A, B, or C) mentioned this specific rule, which is part of the Uniform Commercial Code (UCC), not the general CPLR statute. The LLM has generated this from its parametric memory.
- Entailment Score: 0.15.
- P-value: $p=0.85$, we fail to reject H_0.

Result: Unsupported.

Step 5: Final Output with Multiple Hypothesis Correction

Since three distinct claims were tested, a procedure like the Benjamini-Hochberg (FDR) correction is applied to the p-values to control for false positives. In this case, the decisions remain the same. The final,

validated response presented to the user is constructed only from the supported claims:
"Under New York law, the statute of limitations for a breach of contract is generally six years, as established in NY CPLR § 213. This period begins to run when the cause of action accrues."
This statistically rigorous process prevents the system from presenting information that, while potentially correct, is not grounded in the retrieved evidence, thereby reducing liability and increasing user trust.

VIII. IMPLEMENTATION STRATEGIES AND SCALABILITY CONSIDERATIONS

Integrating hypothesis testing into a production RAG system requires careful consideration of computational cost and latency. The following strategies can help create a scalable and efficient implementation:

A. Pre-computation and Maintenance of Null Distributions:
Instead of constructing the null distribution for retrieval on-the-fly, pre-compute and maintain reference distributions of similarity scores.
These can be created offline for different domains or query types (e.g., "keyword-based," "semantic," "question"). Periodically update these distributions to prevent concept drift. This transforms the p-value calculation into a simple lookup, significantly reducing latency.

B. Tiered and Asynchronous Validation:
Not all checks need to be synchronous. For retrieval, use a fast, heuristic-based top-k selection (e.g., ANN search) to get an initial set of candidates. Then, run the hypothesis test as a rapid, secondary filtering stage.
For response validation, the system can stream the initial, unverified answer to the user immediately, while running the computationally intensive entailment tests asynchronously. Supported claims can then be visually confirmed (e.g., with a checkmark icon) or unsupported ones flagged or retracted.

C. Caching and Memorization:
Cache the results of hypothesis tests for frequently retrieved documents or commonly generated claims. If the same document is retrieved for a similar query, its relevance p-value can be reused. Similarly, entailment scores for common factual statements can be stored to avoid redundant NLI model inferences.

D. Optimized Entailment Models:
Use distilled or quantized versions of NLI models for the response validation stage. While larger models may be more accurate, smaller, fine-tuned models can offer a much better trade-off between performance and speed, which is critical for real-time applications.

E. Adaptive Significance Levels (a):
The stringency of the tests should not be static. Implement a mechanism to tune the significance level (a) based on the application's risk tolerance. For a high-stakes legal or medical query, a very low a (e.g., 0.01) is appropriate. For a low-risk creative writing assistant, a higher a (e.g., 0.10) might be acceptable to allow for more diverse results. This level could even be adjusted based on user feedback or downstream task performance.

F. Managing Computational Overhead:
The primary overhead comes from NLI model inference for every generated claim. To manage this, focus validation on "high-risk" claims, such as those containing numerical data, named entities, or specific legal/medical terminology. Simpler, connective sentences may not require statistical validation. This selective approach balances rigor with performance.
By employing these strategies, the statistical validation layer can be integrated without making the RAG system prohibitively slow, ensuring both reliability and a good user experience.

IX. FUTURE DIRECTIONS

Several advanced integrations are possible:
- Bayesian Hypothesis Testing: Rather than fixed a, use Bayesian priors and compute posterior probabilities of relevance.
- Sequential Testing: As new documents arrive, update hypotheses incrementally.
- RI-Tuned Significance Levels: Use reinforcement learning to adapt a based on reward signals from user satisfaction or task success.
- Causal Hypothesis Testing: Combine hypothesis testing with causal inference to answer counterfactual questions ("What if this document were not included?").

X. CONCLUSION

Hypothesis testing adds a critical layer of statistical inference to Retrieval-Augmented Generation pipelines. It transforms opaque retrieval and generation scores into interpretable, probabilistically meaningful decisions. Whether filtering noisy documents, validating generated claims, or adjusting thresholds dynamically, the use of hypothesis testing in RAG strengthens system robustness, fidelity, and trustworthiness. Future extensions, including Bayesian and causal frameworks, can further enhance their applicability in real-world Al systems.

REFERENCES

[1] Vaswani, A., et al. (2017). "Attention is all you need." Advances in neural information processing systems, 30.

[2] Lewis, P., et al. (2020). "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks." Advances in Neural Information Processing Systems, 33, pp. 9459-9474.

[3] Shuster, K., et al. (2021). "Retrieval Augmentation Reduces Hallucination in Conversation." Findings of the Association for Computational Linguistics: EMNLP 2021, pp. 413-421.

[4] Neyman, J., & Pearson, E. S. (1933). "On the Problem of the Most Efficient Tests of Statistical Hypotheses." Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character, 231(694-706), pp. 289-337.

[5] Laber, E. B., et al. (2014). "A tutorial on bootstrap and permutation testing." Journal of the Brazilian Computer Society, 20(1), pp. 1-14.

[6] Efron, B., & Tibshirani, R. J. (1994). An introduction to the bootstrap. CRC press.

[7] Min, S., et al. (2023). "FActScore: Fine-grained Atomic Evaluation of Factual Precision in Long Form Text Generation." Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pp. 11345-11363.

[8] Bowman, S. R., et al. (2015). "A large annotated corpus for learning natural language inference." Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 632-642.

[9] Benjamini, Y., & Hochberg, Y. (1995). "Controlling the false discovery rate: a practical and powerful approach to multiple testing." Journal of the Royal Statistical Society: Series B (Methodological), 57(1), pp. 289-300.