# SELF-EVALUATION AND SELF-PROMPTING TO IMPROVE THE RELIABILITY OF LLMS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

In order to safely deploy Large Language Models (LLMs), they must be capable of dynamically adapting their behavior based on their level of knowledge and uncertainty associated with specific topics. This adaptive behavior, which we refer to as *self-restraint*, is non-trivial to teach since it depends on the internal knowledge of an LLM. By default, LLMs are trained to maximize the next token likelihood, which does not teach the model to modulate its answer based on its level of uncertainty. In order to learn self-restraint, we devise a simple objective that can encourage the model to produce responses that it is confident in. To optimize this objective, we introduce ReSearch, an iterative search algorithm based on self-evaluation and self-prompting. Our method results in fewer *hallucinations* overall, both for known and unknown topics, as the model learns to selectively restrain itself. In addition, our method elegantly incorporates the ability to *decline*, when the model assesses that it cannot provide a response without a high proportion of hallucination. While ReSearch is expensive, we demonstrate that we can amortize the results of the search and improve the reliability of the models at no additional inference cost.

## 1 INTRODUCTION

In order for Large Language Models (LLMs) to become reliable tools, it is important for the models to be able to modulate their responses based on their internal knowledge. In cases where the models are queried about a topic that is not well supported by their internal knowledge, it is safer for the LLMs to provide a short answer or to even refrain from answering entirely, instead of providing an answer filled with inaccuracies (*hallucinations*). Unfortunately it is non-trivial to teach this behavior to LLMs since the optimal behavior depends on their internal knowledge (Goldberg, 2023).

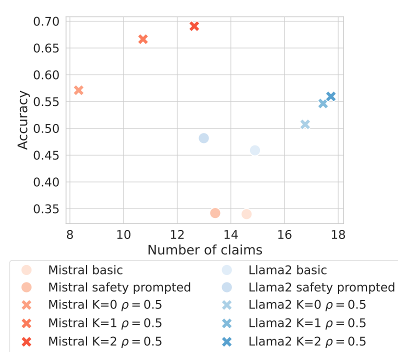


Figure 1: **Accuracy and average number of claims as a function of rewriting rounds (K).** We observe that our search procedure improves upon the *basic* and *safety-prompted* (for specific prompts see Table 1) base models in terms of factual accuracy. In addition to achieving higher accuracy, we observe that our method increases the number of claims for Llama2 7b chat. Increasing the number of ReSearch iterations (K) further improves the accuracy and the number of claims. Generation iterations can be observed into more details in Table 2.

There has been several successful attempts to improve the factuality of LLMs while maintaining their helpfulness using preferences (Tian et al., 2023a) or question-and-answer-based rewriting (Dhuliawala et al., 2023). While these methods have been shown to reduce hallucinations on average, they do not fulfill the desiderata of teaching LLMs to a) reduce the level of detail of their responses based on the content of the query, or b) decline queries that they are entirely unfamiliar with or uncertain about.

In this work, we introduce ReSearch, an iterative search algorithm based on self-prompting and self-evaluation (a visual overview is provided in Figure 2). We show that ReSearch can be used to teach LLMs self-restraint resulting in less hallucination on a biography generation task.

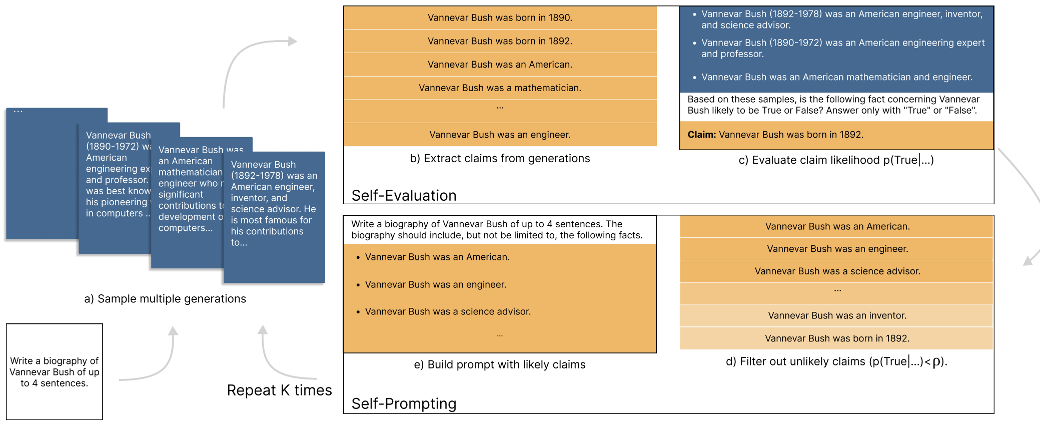


Figure 2: **Illustrative example of ReSearch.** ReSearch combines two components: 1) *Self-Evaluation* where the model evaluate the likelihood of its generated claims based on their self-consistency with all the generations produced by the model, and 2) *Self-Prompting* where the model incorporate the likely claims into its prompt to improve its generations at the next iteration.

## 2 METHOD

We are interested in a model that, given its internal knowledge, exhibits self-restraint i.e., is 1) helpful by generating as many true claims as possible and 2) harmless by limiting the number of false claims it produces. As we show in the experiment section, this behavior is very hard to obtain via prompting. Similarly, building a dataset for this behavior to train an agent via supervised learning is very difficult since it requires us to have access to the model's internal knowledge. Thus, in order to train a model that exhibits self-restraint, we train the model via on-policy reinforcement learning. Using RL requires us to either design a reward function or to learn one from human feedback. We can easily define a reward function that encourages the agent in maximizing the number of true claims and minimizing the number of false claims. We then introduce ReSearch, an iterative search algorithm based on self-evaluation and self-prompting (an overview is provided in Figure 2), to maximize our reward.

### 2.1 REWARD FUNCTION DESIGN

In order to design a reward function, we assume that a generation $y$ can be broken up into atomic claims as shown in Figure 2 b) and that each claim can be judged by an oracle $\mathcal{T}$ as being factual or not. Furthermore, we are interested in having an agent that answers with at least $\rho$ overall accuracy of the claims in its generation and otherwise abstains from answering a query. We define a *factuality reward* $\mathcal{F}(x, y)$ that can be decomposed as a sum over the claims in sample $y$. In order to represent our preference, for a pair of responses with an equal number of false claims, our factuality score must provide a higher reward to the response out of the two that contains more true claims. Since false claims can cause harm, we penalize them more heavily that we reward true claims. Specifically, letting the *oracle* $\mathcal{T}(c)$ be 1 if the claim $c$ is true and 0 otherwise, we obtain

$$\mathcal{F}(x, y) = \sum_{c \in x} R(\mathcal{T}(c)), \tag{1}$$

where

$$R(\mathcal{T}(c)) = \begin{cases} 1 \text{ if } \mathcal{T}(c) = 1 \\ -\lambda \text{ if } \mathcal{T}(c) = 0. \end{cases}$$

We want to select $\lambda$ such that the model must generate a response with an expected factual accuracy of at least $\rho$ or decline to answer otherwise (and obtain a reward of 0). We can solve for $\lambda$, by setting the factuality score equal to 0 for $N$ facts:

$$\mathcal{F}(x, y) = \rho N - \lambda(1 - \rho)N = 0 \tag{2}$$

we can easily see that $\lambda = \frac{\rho}{1-\rho}$. To avoid declining, the model must produce a response with a factual reward of at least 0, as declining to answer yields a reward of 0. Thus the reward can be seen as enforcing a sort of *constrained optimization* where the model must decline to answer if it cannot satisfy the accuracy constraint.

### 2.2 ReSearch: an iterative search algorithm for LLMs

**Self-Evaluation.** To evaluate the factuality reward we outlined previously exactly we must have access to an oracle $\mathcal{T}$, which might not always be possible. Instead, we will rely on an approximate factuality reward model $\hat{\mathcal{F}}(x, y)$. We express the factuality reward as an expectation and leverage self-consistency with the model's responses $Y$, using claim splitter CS:

$$\hat{\mathcal{F}}(x, y) = \sum_{c \in \text{CS}(y)} \sum_{t \in \{0,1\}} R(\mathcal{T} = t) p(\mathcal{T} = t \mid x, c, Y) \tag{3}$$

where $p(\mathcal{T} = 1 | x, c, Y)$ is the probability of a claim being true and is evaluated using $L$ subsets of the generations produced by the model is defined as

$$p(\mathcal{T} | x, c, Y) := \frac{1}{L} \sum_{A_k \subseteq A(Y)} p\big(\texttt{True} \mid \mathcal{P}_{\text{eval}}(x, A_k, c)\big), \tag{4}$$

where $\mathcal{P}_{\text{eval}}$ is a prompt template, `True` is the true token, a set $A(Y)$ of supporting sentences from the model's generation $Y$ selected using BM25 (Amati, 2009), and $A_k$ subsets of the supporting sentences. This self-evaluation method is closely related to Manakul et al. (2023), Kadavath et al. (2022) and Tian et al. (2023b), where the models are shown to improve their calibration by generating and evaluating multiple hypothesis.

**Self-Prompting.** On the first iteration, we initialize the algorithm with a set of likely claims with the empty set $\mathcal{C} = \emptyset$. We sample several model responses $y \sim \pi(\mathcal{P}_{\text{write}}(x))$ from the language model. We then extract likely claims using the self-evaluation framework outlined above and update the set of likely claims with our claim splitter CS: $\mathcal{C} = \{c \text{ for } c \in \text{CS}(y_{\text{best}}) \mid p(c) \geq \rho\}$ [1]. On the following iterations, we sample new generations based on the rewriting prompt including the likely claims set $y \sim \pi(\mathcal{P}_{\text{rewrite}}(x, \mathcal{C}))$. Self-prompting has been used for short form question-answering (Li et al., 2022). See Algorithm 1 for details.

**Amortization.** The ReSearch procedure introduced above is expensive. While it could be used during deployment, it would result in high inference-time latency and cost. Instead, we show that we can amortize the results of the search procedure directly to learn the resulting distribution.

## 3 Experiment

We conduct our experiments using the FActScore task Min et al. (2023). The task consists of asking the model to produce a biography of an individual who has an entry on Wikipedia. The resulting biography is then scored via a larger LLM (LLaMA 2 70B Chat in our case) having access to the text of the Wikipedia entry. Passages are retrieved with GTR-Large (Ni et al., 2021) and injected into the LLM's context. In order to understand the behavior of the models for queries of different difficulties, we expand the FActScore dataset to include 10000 entities, including people that have relatively short Wikipedia entries. People with short Wikipedia articles are likely to be broadly less well known, and as such, there should be less information about them online, and the LLM should know less about them. Therefore we expect in these cases for the model to restrain itself and give a reduced level of detail in its response, or decline entirely. This makes the task significantly more challenging than the set of common public figures that the FActScore paper uses.

**Goal of the experiment.** We examine if 1) 7B models (LLaMA-2 7B Chat (Touvron et al., 2023) and Mistral 7B Instruct v0.1 (Jiang et al., 2023)) can show *self-restraint* and modulate the details of their responses and increase their overall factually accuracy, and 2) the results of the search procedure can be *amortized* into the weight of the LLMs. To investigate these behaviors, we query the model for requests of varying difficulties (with the proxy for difficulty being the corresponding Wikipedia article length) and observe the level of restraint of the model for each difficulty (top, middle and

[1] Note that we make the design choice of only including claims from $y_{\text{best}}$ since it does not require a claims deduplication step.

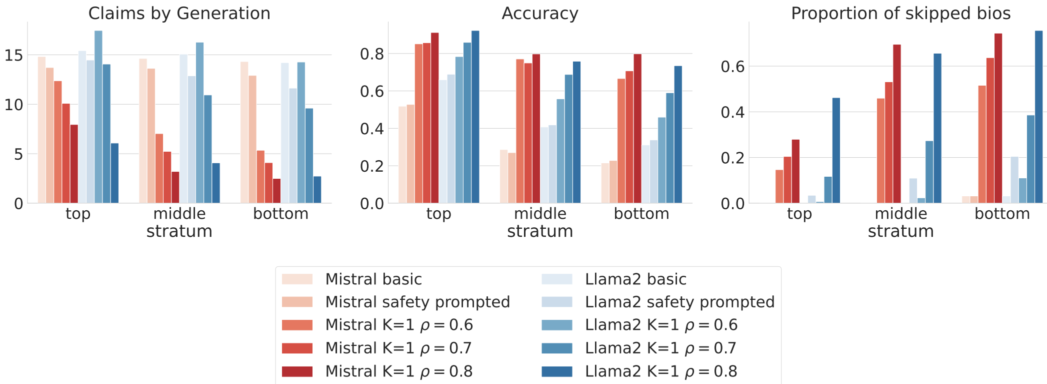


Figure 3: **LLMs behaviors as a function of popularity stratum for different target accuracy $\rho$.** We observe that our method results in LLMs that can modulate their behaviors in term of accuracy, number of claims and skipped bios as a function of the entry popularity (as measure by the length of their Wikipedia). Furthermore, we observe that the basic LLMs and prompted LLMs do not adapt their behavior based on the entry's popularity.

bottom tier entries). We also keep track of the number of claims and the proportion of bios that are declined (skipped), as we would expect both to correlate with the Wikipedia article length (itself a proxy for model knowledge).

**Prompted baselines** are a strong baseline in a variety of tasks. The task of self-restraint is more challenging than other tasks as it requires the model to examine its internal knowledge to modulate its answers. We present two prompted baselines: ordinary greedy decoding from the model (listed as "basic" in all figures), and safety-prompting the model (listed as "safety prompted"). The safety-prompting approach consists of telling the model explicitly in the prompt that it may decline to respond if it is too uncertain. All the prompts can be found in Table 1.

**Trade-off between number of claims and accuracy.** In Figure 3, as expected, we observe that factual accuracy increases as a function of target accuracy $\rho$ (see Section 2.1). We note that the proportion of skipped bios also increases as a function of $\rho$, again following the expected behavior, as well as the number of claims decreasing as $\rho$ is increased, corresponding to a stricter factual accuracy reward and therefore shorter model responses on average. Furthermore, we observe in Figure 1 that increasing the number of rewrites increase the accuracy and the number of claims. Finally, we observe in Figure 4 that the results of the search can be amortized via supervised fine-tuning (SFT) into the weights of Llama2 7b chat, but the amortizing the ReSearch output for Mistral results in more skipped bios and thus in less claims, but higher accuracy.

**Behavior for different strata.** In Figure 3, we observe that the baselines do not modulate the number of claims as a function of the popularity of the entry (as measured by the length of their Wikipedia entry). While ReSearch models reduce the number of claims (and decline to answer) for less popular entities allowing them to maintain higher accuracy for all entities.

## 4 Discussion

In this paper, we explored the ReSearch algorithm's capability to results in LLMs exercising self-restraint. Our findings show that ReSearch can effectively reduce hallucinations in LLM outputs by encouraging the model to modulate its responses or even decline to answer when its knowledge is insufficient. This approach addresses a critical need in the deployment of LLMs, where ensuring the reliability and accuracy of model outputs is paramount. We demonstrate significant improvement compared to both our basic and safety-prompted baselines.

**Limitations and Future Work:** One limitation of our reward function is that it does not take into consideration claim specificity or human usefulness. For example, a model might correctly say that all entries have parents and are human, while true, these claims are not useful. A potential next step would be to further finetune our model with human feedback.

## REFERENCES


Giambattista Amati. *BM25*, pp. 257–260. Springer US, Boston, MA, 2009. ISBN 978-0-387-39940-9. doi: 10.1007/978-0-387-39940-9_921. URL https://doi.org/10.1007/978-0-387-39940-9_921.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. Chain-of-verification reduces hallucination in large language models. *arXiv preprint arXiv:2309.11495*, 2023.

Yoav Goldberg. Reinforcement learning for language models. Retrieved from https://gist.github.com/yoavg/6bff0fecd65950898eba1bb321cfbd81, April 2023. Accessed: February 8, 2024.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

Junlong Li, Zhuosheng Zhang, and Hai Zhao. Self-prompting large language models for zero-shot open-domain qa, 2022.

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

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

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y. Zhao, Yi Luan, Keith B. Hall, Ming-Wei Chang, and Yinfei Yang. Large dual encoders are generalizable retrievers, 2021.

Katherine Tian, Eric Mitchell, Huaxiu Yao, Christopher D Manning, and Chelsea Finn. Fine-tuning language models for factuality. *arXiv preprint arXiv:2311.08401*, 2023a.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D Manning. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. *arXiv preprint arXiv:2305.14975*, 2023b.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

# A Appendix

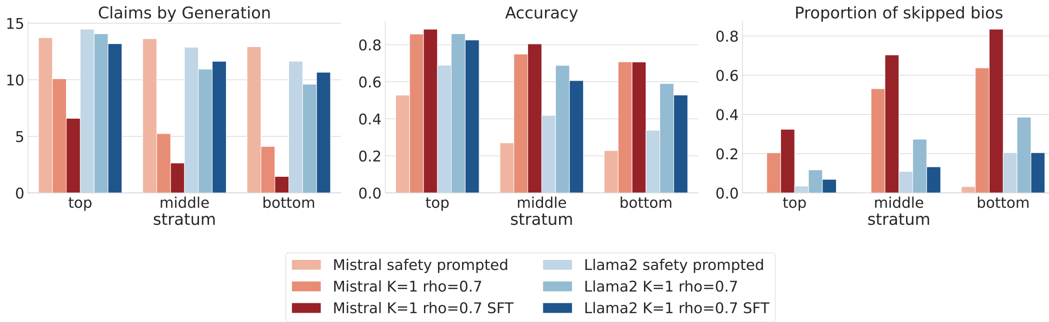


Figure 4: **Amortization.** We observe that the search procedure can be amortized into the weights via supervised fine-tuning (SFT) of Llama2 to a large extend. However amortizing the search procedure into the weights of Mistral results in less facts (and more skipped bios) and slightly higher accuracy.

| symbol | prompt |
|---|---|
| $\mathcal{P}_{\text{write}}$ | `Write a biography of {entity} of up to 4 sentences.` |
| $\mathcal{P}_{\text{safe write}}$ | `Write a biography of {entity} of up to 4 sentences. If you do not know who that is, do not answer.` |
| $\mathcal{P}_{\text{rewrite}}$ | `Write a biography of {entity} of up to 4 sentences. The biography should include, but not be limited to, the following facts.`<br>`Facts:`<br>`{facts}` |
| $\mathcal{P}_{\text{eval}}$ | `...`<br>`{sources} Based on these sources, is the following fact concerning entity likely to be True or False? Answer only with "True" or "False".`<br>`Fact: {fact}` |
| $\mathcal{P}_{\text{splitter}}$ | `...`<br>`Please breakdown the following sentence into independent claims. Each claim must be understandable as an independent sentence, and start explicitly with the name of the entity.`<br>`Entity: {entity}`<br>`Sentence: {sentence}`<br>`Claims:` |

Table 1: **Prompt table.**

**Algorithm 1** ReSearch

**Require:** Context dataset $\mathcal{Q} \leftarrow \{x_i\}_{i=1}^{N}$
**Require:** Policy $\pi_\theta : \mathcal{X} \rightarrow \mathcal{P}(\mathcal{Y})$
**Require:** Factuality reward model $\hat{\mathcal{F}}(x, y) \rightarrow \mathbb{R}$
**Require:** Claim likelihood $p(\mathcal{T}|x, c, Y) \rightarrow [0, 1]$
**Require:** Factuality threshold $\rho \in [0, 1]$
**Require:** Claim splitter $\text{CS}(y)$

```
On-policy dataset 𝒟 ← ∅
for x ∈ 𝒬 do
  Y ← ∅
  # sample J initial generations.
  {y^j ~ π_θ(·|𝒫_write(x))}_{j=1}^J
  while stopping_criterion do
    # Collect generations
    Y ← Y ⋃ {y^j}_{j=1}^J
    # Filter claims
    𝒞 ← {c ∈ CS(Y) | p(𝒯|x, c, Y) > ρ}
    # sample J new generations
    {y^j ~ π_θ(·|𝒫_rewrite(x, 𝒞))}_{j=1}^J
  end while
  # Get the generation with the highest reward
  y_best ← argmax_{y∈Y} F̂(x, y)
  if F̂(x, y_best) ≤ 0 then
    # declining results in reward of 0
    y_best ← decline to answer
  end if
  𝒟 ← 𝒟 ⋃ (x, y_best)
end for
min_θ −𝔼_{(x,y)∼𝒟} log π_θ(y | x)
```

| Prompt | Generation |
|---|---|
| Write a biography of Vannevar Bush of up to 4 sentences. | Vannevar Bush was an American engineer, inventor, and science policy advocate who made significant contributions to the development of modern computing, information management, and artificial intelligence. He is best known for his work on the development of the first automatic electronic computer, the Harvard Mark I, and for his 1945 essay ”As We May Think,” which outlined a vision for a future in which people could access and manipulate information with the touch of a button. Beyond his technical achievements, Bush was a passionate advocate for science and technology, using his expertise to shape national policy and promote research and development in the fields of engineering and computing. - **Expected factuality score: 4.73** |
| Write a biography of Vannevar Bush of up to 4 sentences. The biography should include, but not be limited to, the following facts.<br>Facts:<br>• Vannevar Bush was an engineer.<br>• Vannevar Bush made significant contributions to the development of modern computing.<br>• Vannevar Bush promoted research and development in the fields of engineering and computing.<br>• Vannevar Bush was an American.<br>• . . . | Vannevar Bush was an American engineer and inventor who made significant contributions to the development of modern computing and information management. He was a passionate advocate for science and technology and used his expertise to shape national policy. Bush promoted research and development in the fields of engineering and computing and was a pioneer in the development of computer networks. He is remembered for his visionary ideas and his impact on the evolution of computing and information technology. - **Expected factuality score: 9.48** |
| Write a biography of Vannevar Bush of up to 4 sentences. The biography should include, but not be limited to, the following facts.<br>Facts:<br>• Vannevar Bush was an engineer.<br>• Vannevar Bush is known for his impact on the evolution of computing.<br>• Vannevar Bush made significant contributions to the development of modern computing.<br>• Vannevar Bush promoted research and development in the fields of engineering and computing.<br>• Vannevar Bush is known for his impact on the evolution of information technology.<br>• . . . | Vannevar Bush was an American engineer, inventor, and passionate advocate for science and technology. He is known for his impact on the evolution of computing and his visionary ideas about information technology. Bush made significant contributions to the development of modern computing, including his work on early computer architectures and the concept of the ”memory store.” He also promoted research and development in the fields of engineering and computing, and used his expertise to shape national policy. Despite his many accomplishments, Bush remained a humble and dedicated servant to science and technology throughout his life. Today, he is remembered as a pioneer in the field of computing and a visionary whose ideas have shaped the future of information technology. - **Expected factuality score: 11.71** |

Table 2: ReSearch 2 iterations example.