algorithms

MDPI

Article

Analysis of Reliability and Efficiency of Information Extraction Using AI-Based Chatbot: The More-for-Less Paradox

Eugene Levner [1,*] and Boris Kriheli [2]

[1] School of Computer Science, Faculty of Sciences, Holon Institute of Technology, Holon 58102, Israel
[2] Department of Computer Science, Ashkelon Academic College, Ashkelon 78211, Israel; borisk@hit.ac.il
* Correspondence: levner@hit.ac.il

Abstract

This paper addresses the problem of information extraction using an AI-powered chatbot. The problem concerns searching and extracting relevant information from large databases in response to a human user's query. Expanding the traditional discrete search problem well known in operations research, this problem introduces two players; the first player—an AI chatbot such as ChatGPT 4.0—sequentially scans available datasets to find an appropriate answer to a given query, while the second—a human user—conducts a dialogue with the chatbot and evaluates its answers in each round of the dialogue. The goal of an AI-powered chatbot is to provide maximally useful and accurate information. During a natural language conversation between a human user and an AI, the human user can modify and refine queries until s/he is satisfied with the chatbot's output. We analyze two key characteristics of human–AI interaction: search reliability and efficiency. Search reliability is defined as the ability of a robot to understand user queries and provide correct answers; it is measured by the frequency (probability) of correct answers. Search efficiency of a chatbot indicates how accurate and relevant the information returned by the chatbot is; it is measured by the satisfaction level a human user receives for a correct answer. An AI chatbot must perform a sequence of scans over the given databases and continue searching until the human user declares, in some round, that the target has been found. Assuming that the chatbot is not completely reliable, each database may have to be scanned infinitely often; in this case, the objective of the problem is to determine a search policy for finding the optimal sequence of chatbot scans that maximizes the expected user satisfaction over an infinite time horizon. Along with these results, we found a counterintuitive relationship between AI chatbot reliability and search performance: under sufficiently general conditions, a less reliable AI chatbot may have higher expected search efficiency; this phenomenon aligns with other well-known "more-for-less" paradoxes. Finally, we discussed the underlying mechanism of this paradox.

Keywords: information search; AI chatbot; search reliability; search efficiency; discrete sequential search problem; human–robot search; two-player search problem; more-for-less paradox

check for updates

Academic Editors: Francesc Pozo and Jaroslaw Krzywanski

Received: 17 March 2025
Revised: 11 June 2025
Accepted: 30 June 2025
Published: 3 July 2025

Citation: Levner, E.; Kriheli, B. Analysis of Reliability and Efficiency of Information Extraction Using AI-Based Chatbot: The More-for-Less Paradox. *Algorithms* **2025**, *18*, 412. https://doi.org/10.3390/a18070412

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

This paper addresses the problem of information extraction by an AI-powered chatbot that is not fully reliable and may make mistakes in its responses. The problem concerns searching and extracting relevant information from large datasets in response to a user query. Unlike conventional search and rescue robots that typically perform targeted

searches using well-known pathfinding algorithms to autonomously navigate in physical space, the AI chatbot processes a natural language conversation with a human user and extracts meaningful information through semantic search and human–AI interaction to generate the most relevant and contextually appropriate responses. During a user–AI conversation in natural language, the human user can modify and refine queries until s/he is satisfied with the chatbot's output. We analyzed two key characteristics of human–AI interaction, search reliability and efficiency [1–3]. The reliability of an AI-based chatbot is reflected in its ability to understand user queries and provide correct answers; it is measured by the frequency (probability) of correct responses [1,2]. The search efficiency of a chatbot [3] indicates how accurate and relevant the information returned by the chatbot is. In this study, search efficiency is measured by the level of satisfaction that a human user receives for a correct answer from the chatbot on a 10-point Likert scale. This is a well-known IBM SPSS Statistics version 25 rating system for measuring customer satisfaction, in which the highest score of 10 means "very satisfied" on a 10-point scale from 1 to 10, and the lowest score of 1 means "very dissatisfied" [4].

In this paper, we uncovered a counterintuitive relationship between the reliability of AI chatbots and their search efficiency. We demonstrated that, under general conditions, reducing the reliability of deployed chatbots can improve the search efficiency. In particular, a less reliable chatbot that stops searching after extracting relevant information can have a higher expected efficiency than a more reliable chatbot. This phenomenon aligns with the family of "more-for-less" paradoxes observed in various complex systems. Finally, we discussed the underlying mechanism of this paradox.

The main contributions of this paper were as follows: (i) we presented and described a multi-round conversational process between a human user and an AI chatbot, such as ChatGPT, that served to extract relevant information and consisted of multiple rounds of query modification/refinement during the conversation; (ii) we formulated the information extraction optimization problem for an imperfect AI chatbot as a two-player discrete sequential search problem with probabilistic parameters, where the first player, the AI chatbot, performed the search, while the second player, the human user, evaluated the performance of the first player at each search iteration; (iii) we developed an efficient index-based algorithm for finding the optimal search policy, which extended the well-known fast scheduling algorithms of Ross [5], Sweat [6], and Chew [7] for discrete sequential search problems; and finally, (iv) we analyzed and explained the counterintuitive relationship between the AI chatbot reliability and the search efficiency in the multi-round information search/extraction process.

The remainder of the paper is organized as follows. In the next section, we introduce the main definitions that will be used in the sequel. Section 3 describes the information extraction problem to be solved. Section 4 provides a brief overview of the previous work. In Section 5, we propose a mathematical model and an efficient index-based algorithm to optimize the expected user satisfaction over an infinite horizon. In Section 6, we formulate and prove the paradoxical relationship between the reliability and efficiency of an optimal information extraction process using an AI chatbot. A numerical example to support and illustrate the studied paradox is presented in Section 7. A summary of the contributions and an outline of future research are provided in Section 8.

2. Preliminaries $\sum_i$

In this section, we provide the basic definitions that will be used in the sequel.

2.1. The Discrete Search Problem with Imperfect Robot Sensors

The common *discrete search problem* well known in operations research is as follows [5–10]. A target is hidden in one of the known locations, 1, 2, . . ., N, with a probability distribution $p = (p_1, p_2, \ldots, p_N)$, $\sum_i p_i = 1$, and remains in this location during the search. A search device (for example, the search-and-rescue robot) must perform a sequence of inspections at locations 1, 2, . . ., N to find the target and continue searching until the target is found.

The robot's sensor is *imperfect*, that is, for each location i, it has a non-zero overlook probability a_i associated with it, so that if the target is at location i and the robot is exploring that location, there is a chance (probability $a_i > 0$) that the sensor will not detect the target for any $i = 1, \ldots, N$.

Given the above probabilities, along with the duration of the inspections and the reward for detecting the target at a given location, the goal is to define a probabilistic search policy that finds the optimal sequence of robot movements that maximizes the expected total reward over an infinite time interval. Specific details regarding information search/extraction using AI-powered chatbots are presented in the next subsection.

2.2. The Information Extraction Problem Using AI Chatbots

The studied problem refers to searching and extracting relevant information from a large dataset in response to a user's query. The goal of an AI chatbot is to provide the most useful and accurate information and avoid irrelevant results.

While the task of information extraction using AI chatbots can be considered a special class of the general discrete search problem, these problems differ in their goals, physical environments, and solution methods. Search-and-rescue robots perform searches, typically after disasters, in physical areas, seeking to find and rescue survivors, while AI-powered chatbots perform information search and extraction, aiming to find, extract, and generate relevant textual and visual answers from digital knowledge bases, documents, and the web. Search-and-rescue robots use discrete pathfinding algorithms such as A* and Dijkstra's algorithm to autonomously navigate physical spaces, while the conversation between a human user and an AI chatbot consists of multiple stages (rounds), during which the user modifies/refines queries while the chatbot improves its responses over time based on user feedback and learning from the user's interactions with the AI, until the user is satisfied.

An AI-powered chatbot searches, discovers, and extracts the most relevant information requested through a process called semantic search. This is an advanced information retrieval technology that focuses on understanding the meaning and content of a user's query, rather than simply searching and matching keywords. The chatbot typically scans hundreds of databases and their clusters depending on the complexity of the query, where thousands of documents can be scanned in milliseconds, making this process efficient. The process consists of the following main steps:

- Query understanding, where the chatbot interprets the human user's request using natural language processing to extract the meaning and then convert the text into a numerical format;
- Using semantic search, the chatbot compares the query with data stored in databases or multiple database clusters to find the most relevant matches;
- The chatbot performs ranking/filtering of the query to prioritize the most relevant pieces of information and refine its response;
- Once the relevant information has been found, the robot generates text using natural language and extracts the natural language response.

In the next section, we present a mathematical model of the problem.

3. Description of the Efficiency-Maximizing AI-Based Search Problem

The target information is hidden in one of the known database clusters (called locations) 1, 2, . . ., N, with a known probability distribution $p = (p_1, p_2, \ldots, p_N)$, $\sum_i p_i = 1$, and remains in this location during the search. The search device (in our case, the AI-powered chatbot) must perform a sequence of scans in locations 1, 2, . . ., N to find the target information and continue the multi-round search until a human user declares, on some round, that the target has been found; in this case, the search process stops.

The chatbot is imperfect (not completely reliable), i.e., at each stage, for each location i, it has a non-zero overlook probability ai, so that if the requested target information is in location i and the chatbot scans this location, it does not recognize it and does not report the presence of the target information there, $i = 1, \ldots, N$.

Given the above probabilities $p = (p_1, p_2, \ldots, p_N)$ and $a = (a_1, a_2, \ldots, a_N)$, as well as the duration of all conversation rounds and the user satisfaction levels with the extracted target information in each conversation round, the objective of the information extraction problem under consideration is to determine a stochastic search policy to find the optimal sequence of chatbot location scans that maximizes the expected user satisfaction over an infinite time horizon. In Table 1, we highlight the main differences between the general discrete search problem, well known in operations research, and the information extraction problem. The essential difference between them is that the present work is motivated by and focuses on the practical issue of analyzing the properties of existing AI chatbots.

Table 1. Comparison of two problems.

Feature \ Problem	Discrete Search Problem	AI-Powered Chatbot Information Extraction Problem
Objective function	Expected (discounted) reward over infinite time interval.	Expected efficiency over infinite horizon.
Stopping rule	Search stops at robot's step when the sensor reports that it has detected a hidden target.	Information search stops when the human user is satisfied with the chatbot's result.
Decision to be found	Optimal infinite sequence of robot steps (inspections of the locations).	Optimal infinite sequence of chatbot scans (inspections of digital databases).
Optimal policy π_t^*	The robot's steps are ordered according to the "attractiveness indices" of locations.	Chatbot's scans are ordered according to the "attractiveness indices" of locations, defined in Section 5.

The mathematical model is considered in more detail. The discrete search area A contains N known database clusters (locations), in one of which a target information is hidden, with a known probability distribution $p = (p_1, p_2, \ldots, p_N)$, $\sum_i p_i = 1$. The AI chatbot inspects the area to find and identify a hidden target whose location is unknown in advance; we assume that this smart robot must sequentially perform location scans in each round, one after another. The AI chatbot is imperfect; this means that associated with each location i is a non-zero overlook probability a_i, that is, a_i is the conditional probability that the robot will not detect the target information in a single scan of location i, given that this location in fact contains the requested information, $i \in A = \{1, \ldots, N\}$. Therefore, since $a_i \neq 0$, for the search to be successful, each location must be searched infinitely often, and, from the mathematical point of view, in general, the sequence of user–chatbot conversational rounds can be infinite.

The duration of the user's query modifications at each conversation round are assumed to be known in advance. Also, as we indicated above, in this work, the search efficiency is measured by the satisfaction level that a human user feels whenever he/she is satisfied

by the chatbot response during some stage (round) of the conversation; it is assumed to decrease monotonically in time during the user's conversation with the chatbot.

The goal of information extraction is to determine the optimal search strategy to find the target information with the maximum expected search efficiency (user's satisfaction) over an infinite time horizon.

In addition to probabilities p_i and a_i defined above, we used the following notations:

- Policy $\boldsymbol{\pi}$ is an infinite sequence of conversation stages (rounds), $\boldsymbol{\pi} = (\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \ldots)$; $\boldsymbol{\pi}_k$ is the number of a location (a database cluster) scanned at the kth round of the sequence $\boldsymbol{\pi}$, $\boldsymbol{\pi}_k \in A = \{1, \ldots, N\}$, $k = 1, 2, \ldots$;
- n is the (unknown) round of the sequence $\boldsymbol{\pi}$ at which the chatbot successfully extracts the targeted information;
- $u_i = u(\boldsymbol{\pi}, n)$ is the number of unsuccessful scans (rounds) of location $i = \boldsymbol{\pi}_n$ before the nth (successful) round of the sequence $\boldsymbol{\pi}$; $u(\boldsymbol{\pi}, n) \leq n - 1$;
- $P_{\boldsymbol{\pi}_n}$ is the conditional probability that after $u(\boldsymbol{\pi}, n)$ unsuccessful scans of location $\boldsymbol{\pi}_n$, the chatbot will successfully detect that location (database cluster); $\boldsymbol{\pi}_n$ contains the target information, given that the latter information is at location $\boldsymbol{\pi}_n$; $P_{\boldsymbol{\pi}_n} = a_{\boldsymbol{\pi}_n}^{u(\boldsymbol{\pi},n)}(1 - a_{\boldsymbol{\pi}_n})$;
- $\Pi_{\boldsymbol{\pi}_n} = P_{\boldsymbol{\pi}_n} p_{\boldsymbol{\pi}_n}$ is the unconditional probability that the chatbot will find the requested information at location $\boldsymbol{\pi}_n$;
- t_k is the known time required for a human user to read, evaluate, and refine the chatbot's response in round k, $k = 1,2, \ldots$;
- $T_{\boldsymbol{\pi},n} = \sum_{i=1}^{n} t_{\boldsymbol{\pi}_i}$ is the total time spent by a human user in conversation with the chatbot from round 1 to round n in sequence $\boldsymbol{\pi}$;
- $r(\boldsymbol{\pi}) = r_{\boldsymbol{\pi}_n}$ is the known Likert-scale satisfaction level obtained by the human user when the target information is found at location $\boldsymbol{\pi}_n$ of the sequence $\boldsymbol{\pi}$ (before discounting). If the search for the target information requires $T_{\boldsymbol{\pi},n}$ time units, the level of satisfaction is discounted by $e^{-dT_{\boldsymbol{\pi},n}}$ for a known discount factor d, $d \in (0,1)$, i.e., as defined as follows: $R_{discount}(\boldsymbol{\pi}) = r_{\boldsymbol{\pi}_n} e^{-dT_{\boldsymbol{\pi},n}} = r_{\boldsymbol{\pi}_n} e^{-d\sum_{i=1}^{n} t_{\boldsymbol{\pi}_i}}$.

In the problem under consideration, the satisfaction level is a random variable with the value $x = r_{\boldsymbol{\pi}_n} e^{-d\sum_{i=1}^{n} t_{\boldsymbol{\pi}_i}}$ and the probability $p(x) = P_{\boldsymbol{\pi}_n} p_{\boldsymbol{\pi}_n} = a_{\boldsymbol{\pi}_n}^{u(\boldsymbol{\pi},n)}(1 - a\boldsymbol{\pi}_n)\, p_{\boldsymbol{\pi}_n}$, depending on $\boldsymbol{\pi}$. Then, the expected discounted satisfaction (search efficiency) corresponds to the expected accumulated information obtained by the human user in the policy $\boldsymbol{\pi}$, denoted as $F(\boldsymbol{\pi})$, and will be defined over an infinite time horizon as follows:

$$F(\boldsymbol{\pi}) = E(R_{discount}\ (\boldsymbol{\pi})) = E(r_{\boldsymbol{\pi}_n} e^{-d\sum_{i=1}^{n} t_{\boldsymbol{\pi}_i}}) = \sum_{n=1}^{\infty} r_{\boldsymbol{\pi}_n} \cdot e^{-d\sum_{i=1}^{n} t\boldsymbol{\pi}_i} \cdot P_{\boldsymbol{\pi}_n} p_{\boldsymbol{\pi}_n} = \sum_{n=1}^{\infty} r_{\boldsymbol{\pi}_n} \cdot e^{-d\sum_{i=1}^{n} t_{\boldsymbol{\pi}_i}} \cdot a_{\boldsymbol{\pi}_n}^{u(\boldsymbol{\pi},n)}(1 - a_{\boldsymbol{\pi}_n}) p_{\boldsymbol{\pi}_n}. \quad (1)$$

The goal is to define a stochastic policy for a search chatbot that determines where it should search at each round, that is, it determines an optimal infinite sequence of chatbot actions that maximizes the expected discounted satisfaction of a human user F($\boldsymbol{\pi}$) over an infinite period of time.

4. Previous Work

This section is devoted to two different topics of this study: a review of efficient discrete search algorithms and a brief overview of paradoxical phenomena arising in operations research.

4.1. Efficient Algorithms for the Discrete Sequential Search Problems

The discrete search problem is one of the oldest problems of operations research, first formulated and solved by Bernard Koopman and his team during World War II, when the military task was to provide effective methods of detecting submarines [11]. An excellent

review of search theory and a recent bibliography of the discrete search literature can be found in [8,12,13].

Several authors have studied discrete sequential search problems with overlooking probabilities in various settings and developed efficient index-based solution methods (see [5–10]). When a search device, such as a search-and-rescue robot or an unmanned aerial vehicle (drone), is not reliable, the inspection/search of a location is imperfect, and in this case, any location may need to be inspected many, even infinite, times. Thus, the discrete search problem is to find an optimal strategy rather than the optimal search sequence. The latter optimal strategy depends on the random moment when the search device finds the target.

A typical stopping rule for a multi-step stochastic search process is: Stop at a step when the target is found or when the assigned search time has elapsed. Other criteria for terminating the search process, called stopping rules, have been proposed by Ross [5] and Chew [7]. Trummel and Weisinger [14] and Wegener [15] showed that the general discrete search problem is NP-hard.

In a more recent study, Kriheli et al. [16] proposed a fast algorithm for searching for hidden targets by unmanned aerial vehicles under imperfect inspections with non-zero, false-positive, and false-negative detection probabilities.

Kress et al. [17] analyzed several myopic and biology-inspired search strategies that minimize search time and showed that a local index-based policy is optimal when target detection by an autonomous automated search device is accompanied by reliable verification work performed by a team of humans.

The discrete sequential search problems mentioned above are single-player search problems in which the decision-maker must optimize a single objective function, such as maximizing the expected reward or minimizing the expected cost, given stochastic or deterministic constraints imposed by the environment; typically, it is necessary to find a single optimal alternative from a finite or infinite set of alternatives. In contrast, two-player search problems are usually studied and solved in the context of game theory as zero-sum games, in which two players compete to find a hidden object or location, with or without cooperation from each other.

The two-player game involves a seeker who moves through space and a hider who tries to prevent the seeker from finding the hidden object. The presence of a second rational player radically changes the problem from optimization to forecasting and combating the optimal counter strategies of another player. This introduces fundamentally new methods, moving from search theory to game theory, where the optimal decisions of each player are chosen depending on the decisions of a rational opponent or a cooperative player [18,19].

In recent years, two-player game problems and their solution methods have become a viable and widely used tool for analyzing and optimizing human–robot interactions. It is a new, rapidly growing area of research in artificial intelligence that studies efficient interactions between humans and robots, particularly through the lens of their joint performance of complex tasks, with direct interaction and coordination [20,21]. Importantly, searches involving human–robot interaction generally outperform hidden object searches by an autonomous robot [22,23].

In this paper, we continue and extend models of human–robot interaction by focusing on the study of a multi-round search scenario in which imperfect inspections of given databases are performed by an AI-powered chatbot, accompanied by a human user who verifies the chatbot's responses and refines his/her queries.

4.2. Paradoxes in Operations Research

In teaching operations research and artificial intelligence courses, the authors have found that presenting counterintuitive or paradox-like anomalies can be a great eye-opener for listeners. Studying such anomalies leads to a better understanding of algorithms and a deeper learning of operations research and AI techniques. A typical example is the famous Braess paradox that arises in dynamic transportation networks and states that "*adding a new wide road to a congested transport network can lead to an increase in overall travel time*" [24,25]. Similar counterintuitive effects are known to also occur, for instance, in power transmission systems [26], biological and ecological systems [27], and exploiting generative artificial intelligence [28].

A mechanism of this paradox can be explained by changing drivers' priorities and behaviors after a network's reconstruction. For example, in response to a capacity addition, more drivers begin to use the expanded highway; those previously not traveling at peak times shift to the peak hours, and public transport users shift to driving. From a mathematical perspective, such drivers' behaviors disrupt the equilibrium and lead to suboptimal traffic flow for everyone. This is because the Nash equilibrium of such a system is not necessarily optimal. The network change induces a new game structure between multiple drivers, which leads to a new situation. Unlike a Nash equilibrium, where drivers have no incentive to change their routes, when the system is not in a Nash equilibrium, individual drivers can improve their respective travel times by changing the routes they take. It follows that drivers will continue to switch until they reach a new Nash equilibrium, despite the reduction in overall performance.

Another phenomenon called the Jevons paradox describes a situation where technological progress increases the efficiency of resource use, reducing the amount required per user and lowering its cost. However, the falling cost of use increases demand. Consequently, while individual resource consumption decreases, total societal resource use by all users increases, regardless of whether this resource is fuel, a computer, or an electronic device. Therefore, this "paradox" is explained by the substantial societal increase in demand, which outweighs individual gains in resource efficiency, leading to a net increase (rather than decrease) in societal resource consumption [29,30]. These paradoxes warn against making decisions based solely on local or individual optimization without assessing the broader impact on the system.

One more paradox arising in scheduling theory was described by Spieksma and Woeginger [31] and states that "*increasing the speed of some machines in a no-wait flow-shop problem can significantly worsen the optimal total time*". There are other, less well-known anomalies studied in literature, for example, the so-called "*more-for-less paradox*" in the classical transportation problem [32] and Graham's multiprocessor scheduling anomaly [33].

The next sections present another paradoxical "more-for-less" phenomenon, according to which increasing generative AI's reliability decreases its efficiency. Note that in all mentioned papers and in this one, the word "paradox" is used to denote a counterintuitive and surprising phenomenon, not a classical paradox in the context of formal logic.

5. Finding the Optimal Policy π^* for the Problem

In this section, we proved the optimality of the greedy index policy for the information extraction problem. We used a basic method called the "interchange argument", which involves exchanging two consecutive search locations. For completeness, we described this index policy for solving the problem under study. In what follows, we used this policy to identify the paradoxical property of the problem. The idea of this method was as follows.

To find the optimal search strategy that maximizes the expected discounted satisfaction presented in (1), it sufficed to compare the objective function $F(\boldsymbol{\pi})$ for two conjugate infinite

sequences $\boldsymbol{\pi}^1$ and $\boldsymbol{\pi}^2$, in which exactly two arbitrary neighboring locations, $\boldsymbol{\pi}^1_n$ and $\boldsymbol{\pi}^1_{n+1}$, were swapped:

$$\boldsymbol{\pi}^1 = \left(\boldsymbol{\pi}^1_1, \boldsymbol{\pi}^1_2, \ldots, \boldsymbol{\pi}^1_{n-1}, \boldsymbol{\pi}^1_n, \boldsymbol{\pi}^1_{n+1}, \boldsymbol{\pi}^1_n, \boldsymbol{\pi}^1_{n+2}, \ldots\right);$$
$$\boldsymbol{\pi}^2 = \left(\boldsymbol{\pi}^1_1, \boldsymbol{\pi}^1_2, \ldots, \boldsymbol{\pi}^1_{n-1}, \boldsymbol{\pi}^2_n = \boldsymbol{\pi}^1_{n+1}, \boldsymbol{\pi}^2_{n+1} = \boldsymbol{\pi}^1_n, \boldsymbol{\pi}^2_{n+2} = \boldsymbol{\pi}^1_{n+2}, \ldots\right).$$

Having performed elementary arithmetic operations with expression (1), it was easy to obtain the following:

$$\begin{gathered} F\left(\boldsymbol{\pi}^1\right) - F\left(\boldsymbol{\pi}^2\right) = \\ r_{\boldsymbol{\pi}^1_n} \cdot e^{-dT_{\boldsymbol{\pi}^1,n}} \cdot P_{\boldsymbol{\pi}^1_n} p_{\boldsymbol{\pi}^1_n} + r_{\boldsymbol{\pi}^1_{n+1}} \cdot e^{-dT_{\boldsymbol{\pi}^1,(n+1)}} \cdot P_{\boldsymbol{\pi}^1_{n+1}} p_{\boldsymbol{\pi}^1_{n+1}} - \\ r_{\boldsymbol{\pi}^1_{n+1}} \cdot e^{-d\left(T_{(\boldsymbol{\pi}^1(n+1))} - t_{\boldsymbol{\pi}^1_n}\right)} \cdot P_{\boldsymbol{\pi}^1_{n+1}} p_{\boldsymbol{\pi}^1_{n+1}} - r_{\boldsymbol{\pi}^1_n} \cdot e^{-dT_{(\boldsymbol{\pi}^1,n+1)}} \cdot P_{\boldsymbol{\pi}^1_n} p_{\boldsymbol{\pi}^1_n}. \end{gathered} \tag{2}$$

From (2), it followed that $F(\boldsymbol{\pi}^1) \geq F(\boldsymbol{\pi}^2)$ if:

$$\frac{r_{\boldsymbol{\pi}^1_n} \cdot e^{-dt_{\boldsymbol{\pi}^1_n}} \cdot P_{\boldsymbol{\pi}^1_n} p_{\boldsymbol{\pi}^1_n}}{1 - e^{-dt_{\boldsymbol{\pi}^1_n}}} \geq \frac{r_{\boldsymbol{\pi}^1_{(n+1)}} \cdot e^{-dt_{\boldsymbol{\pi}^1_{(n+1)}}} \cdot P_{\boldsymbol{\pi}^1_{n+1}} p_{\boldsymbol{\pi}^1_{(n+1)}}}{1 - e^{-dt_{\boldsymbol{\pi}^1_{(n+1)}}}},$$

or

$$\begin{gathered} \frac{r_{\boldsymbol{\pi}^1_n} \cdot e^{-dt_{\boldsymbol{\pi}^1_n}} \cdot a_{\boldsymbol{\pi}^1_n}^{u(\boldsymbol{\pi}^1,n)} \left(1 - a_{\boldsymbol{\pi}^1_n}\right) p_{\boldsymbol{\pi}^1_n}}{1 - e^{-dt_{\boldsymbol{\pi}^1_n}}} \geq \\ \frac{r_{\boldsymbol{\pi}^1_{(n+1)}} \cdot e^{-dt_{\boldsymbol{\pi}^1_{(n+1)}}} \cdot a_{\boldsymbol{\pi}^1_{(n+1)}}^{u(\boldsymbol{\pi}^1,n+1)} \left(1 - a_{\boldsymbol{\pi}^1_{(n+1)}}\right) p_{\boldsymbol{\pi}^1_{(n+1)}}}{1 - e^{-dt_{\boldsymbol{\pi}^1_{(n+1)}}}}. \end{gathered} \tag{3}$$

Since neighboring locations were chosen arbitrarily, in order for a sequence $\boldsymbol{\pi}^*$ to have a maximum expected efficiency $F(\boldsymbol{\pi}^*)$, it was sufficient that all of its components were chosen sequentially one after another in a non-decreasing order of the coefficients defined in (3). We came to the conclusion that the following two-index parameter $Q_{i,j}$ characterized the "attractiveness" of location i, given that this location was already searched unsuccessfully j times, $j \geq 1$:

$$Q_{i,j} = \frac{r_i e^{-dt_i} \cdot a_i^j (1 - a_i) p_i}{1 - e^{-dt_i}}, \; i = 1, \ldots, N; j = 1, 2, \tag{4}$$

Remark. *The symbol u($\boldsymbol{\pi}$,n) in Formula (3) denotes the same term as the symbol j in Formula (4), i.e., the number of times a certain location i = $\boldsymbol{\pi}_n$ has already been unsuccessfully scanned by an unreliable chatbot. However, introducing such a double notation in this context was performed for convenience in order to write the formulas in a more readable form. The former notation was convenient when comparing two policies denoted as $\boldsymbol{\pi}^1$ and $\boldsymbol{\pi}^2$, while the latter will be more convenient below when describing the upcoming multi-round algorithm.*

Using relations (2)–(3) and notation (4), we obtained the following result. The policy which maximized the expected search efficiency (i.e., user satisfaction) was defined as follows:

Proposition 1. *If there have already been j = j(i) unsuccessful scans of each location i, where i ∈ {1, …, N} and j ≥ 1, then choosing for the next scan the location i*, which has the maximum value of the parameter $Q_{i,j}$ defined in (4) among all i for a fixed j = j(i), ensures the maximum discounted satisfaction F($\boldsymbol{\pi}$) of the user.*

5.1. Iterative Algorithm for Constructing the Optimal Sequence

Proposition 1 formulated above allowed us to construct an efficient iterative algorithm for constructing an optimal search policy $\boldsymbol{\pi}^*$.

The algorithm was defined for an indefinite number of search rounds over an infinite time horizon. It contained an unknown number of rounds and continued until the human user was satisfied with the information obtained; in this case, the user stopped the search process. In essence, this procedure requires that at each round, for each subsequent search, exactly one known cluster of databases is selected and scanned, namely the cluster with the maximum parameter $Q_{i,j}$, defined in (4), among the values of $Q_{i,j}$ that have not yet been selected.

The results of the calculations at any round k, $k = 1,2, \ldots$ are represented as a $k \times N$ matrix, denoted as M_k, in which N columns correspond to predetermined locations for scanning, and the rows correspond to the search rounds performed before the algorithm stops. The elements $m_{i,j}$ of the matrix M_k represent the parameters $Q_{i,j}$ from expression (4), computed sequentially round by round. Below, we describe round 1 separately, since it differed from all subsequent rounds in that it contained an additional step, designated as 1.1, in which the first row of the matrix M_1 was formed. After that, we offer a description of the current kth round, the same for all $k \geq 2$.

As mentioned, in this procedure, the subscript j denotes how many unsuccessful scans of location i ($i = 1, \ldots, N$) were made by the chatbot before the start of round k ($k = 1, 2, \ldots$). Therefore, this index can be provided with two indicators, namely i and k, and the subscript j was replaced by the notation $j_k(i)$.

5.2. Rounds of the Algorithm

Let us first describe round 1, which consisted of the following four steps.

Step 1.1. Calculate the elements of the first row of the matrix M_1.

In accordance with expression (4), we set $j = 1$ for all locations, calculated the parameters $Q_{i,1}$ for all $i = 1, \ldots, N$, and wrote them as the corresponding elements $m_{i,1}$ of the first row of the matrix M_1.

Step 1.2. The chatbot performed the first (preliminary) scan of each location one by one and extracted the relevant information from each location. With the chatbot being unreliable, the human user verified that the target information from the scanned-once locations was not satisfiable yet and hence assigned 1 to the number of unsuccessful checks for each location.

Step 1.3. Determine the location i_1^* and scan it once more.

Location i_1^* was chosen such that $Q_{i_1^*,1} = max_i\, Q_{i,1}$. Location $i_1^* = \boldsymbol{\pi}_1^*$ (for the second time) was scanned. If the human user was satisfied with the target information extracted now from the location $\boldsymbol{\pi}_1^* = i_1^*$, the partial satisfaction level $F(\boldsymbol{\pi}^*)$ obtained so far was computed, where $\boldsymbol{\pi}^* = \{\boldsymbol{\pi}_1^*\}$, and the search was terminated; $Q_{i_1^*,1}$ in cell $(1, i_1^*)$ was highlighted in bold (see numerical example in Section 7). Otherwise, the number of unsuccessful scans for location i_1^* was increased by 1, i.e., set $j_2(i_1^*) := j_1(i_1^*) + 1 = 2$. Go to the next round, round 2.

Let us now assume that the rounds $1, \ldots, k$ ($k \geq 1$) were completed, and the locations $\boldsymbol{\pi}^*_1, \ldots, \boldsymbol{\pi}^*_k$ were determined at these rounds, i.e., the optimal subsequence $\boldsymbol{\pi}_{\boldsymbol{k}}^* = (\boldsymbol{\pi}_1^*, \boldsymbol{\pi}_2^*, \ldots, \boldsymbol{\pi}_k^*)$ was found. At the end of the kth round, all parameters $Q_{i,j}$, $i = 1, \ldots, N$, $j = 1,2, \ldots$, and $k + 1$ were calculated and represented as a known $(k + 1) \times N$-matrix $M_{(k+1)}$. Accordingly, the corresponding indices $j_1(i_1{}^*)$, $j_2(i_2{}^*)$, …, and $j_k(i_k{}^*)$ and the objective function $F(\boldsymbol{\pi}^*)$ were calculated. Then, the next, $(k + 1)$th round was as follows.

In round $k + 1$, $k = 1, 2, \ldots$, consisted of the following three steps.

Step $k + 1.1$. Determine the location $i^*_{(k+1)}$ to be scanned $(k + 1)$th-in-turn by the iterative algorithm.

The second index j in the elements of the $(k + 1)$th row of the matrix M_{k+1} by $j_{k+1}(i)$ for all i was determined; among all parameters $Q_{i,j_{k+1}(i)}$, the location i^*_{k+1} (to be scanned as the next one) was selected, which had the maximum $Q_{i,j_{k+1}(i)}$ as follows: $Q_{i^*_{k+1},\, j_{k+1}} = max_{i=1,\ldots,N}\, Q_{i,j_{k+1}(i)}$. The item $Q_{i^*_{k+1},\, j_{k+1}}$ was placed in cell $(k + 1, i^*_{k+1})$ of the $(k + 1)$th row of the matrix M_{k+1}, and this item was highlighted in bold.

Step $k + 1.2$. The human user checked whether the location i^*_{k+1} contained the required target information; if yes, the search stopped; otherwise, the number of failed checks for location i^*_{k+1} was increased by 1.

A current optimal subsequence $\boldsymbol{\pi}^* = \boldsymbol{\pi}^*_{k+1}$ was considered, which, for now, was $\boldsymbol{\pi}^*_{\boldsymbol{be}k+1} = (\boldsymbol{\pi}^*_1, \boldsymbol{\pi}^*_2 \ldots, \boldsymbol{\pi}^*_{k+1})$; the efficiency $F(\boldsymbol{\pi}^*_{k+1})$ was computed. The number of unsuccessful scans of location $\boldsymbol{\pi}^*_{k+1}$ increased by 1, that is, the second index for location $\boldsymbol{\pi}^*_{k+1}$ was changed as follows: $j_{k+2}(\boldsymbol{\pi}^*_{k+1}) := j_{k+1}(\boldsymbol{\pi}^*_{k+1}) + 1$, leaving the second indices for all other locations unchanged: $j_{i,k+2} := j_{i,k+1}$, $i \neq i^*_{k+1}$. Go to step $k + 1.3$.

Step $(k + 1).3$. Determine all elements $Q_{i,j}$ in the $(k + 2)$th row of matrix M_{k+2}.

The $(k + 2)$th row of the matrix $M_{(k+2)}$, in which the elements $Q_{i,j}$ were calculated, was defined as follows: (i) the parameter $Q_{i^*_{(k+2)},\, j_{(k+2)}}$ was placed in cell $(k + 2, j_{k+2}(\boldsymbol{\pi}^*_{k+1}))$, and (ii) all other elements of the $(k + 2)$th row of $M_{(k+2)}$ must be the same as the corresponding elements in the previous row $k + 1$ in $M_{(k+1)}$, that is, $Q_{i,j_{k+2}(i)} := Q_{i,j_{k+1}(i)}$, for $i \neq i^*_{k+1}$. Go to the next round. Rows $1, \ldots, k + 1$ of the matrix M_{k+2} were left the same as the rows in the matrix M_{k+1}; thus, the matrix M_{k+2} was built. The rounds were repeated until a human user was satisfied with the target information extracted by the chatbot.

We concluded this section by discussing the complexity of the proposed algorithm. First, we noted that the sequential scan of datasets (also called *full scan*), which was considered in this paper, was the simplest (and at the same time the slowest) search procedure compared to indexed and vector search, widely used by AI chatbots. However, sequential search is often used in cases where there is no index on the queried field, for example, if the data is loaded on-the-fly without a search index.

The worst-case complexity C (running time) of the greedy iterative algorithm was defined as the sum of the chatbot's execution time $T_{chatbot}$ and the response time T_{user} of the human user, determined at each round of the algorithm and multiplied by the total number of available rounds R of the chatbot–human dialogue, that is, $C = (T_{chatbot} + T_{user})$R, where $T_{chatbot} = N \cdot t_{scan}$; N is the total number of locations, and t_{scan} denotes the average scanning time of one location. The AI chatbot was typically expected to produce one scan within 100–200 milliseconds; T_{user} is usually a few minutes at each round. The number of the rounds was not known *a priori*, before the dialogue was completed, but it depended on the user's satisfaction or the entire time budget allocated in advance. Thus, the total running time of the algorithm was proportional to the size $\mathrm{R} \times N$ of the current matrix M_{R} computed for R rounds of the algorithm.

The proposed iterative procedure is illustrated by a numerical example in Section 7. It shows the operation of the algorithm and demonstrates that the paradoxical property manifested itself already at the early rounds of the considered iterative search process.

6. Comparison of Two Chatbots: The "Paradox"

The goal of this section was to uncover a counterintuitive property of the general AI chatbot information extraction problem. When using optimal search policy, an AI based chatbot must find and extract target information in a multi-round iterative process. We supposed that since the search chatbot was unreliable, the human user was never satisfied with the chatbot's first answer and asked the latter to improve it with a refined query; in other words, a successful round was always preceded by at least one unsuccessful round, i.e., we assumed that the parameter $u(i)$ exceeded 1 for any i.

Suppose we have two search chatbots with the same time/efficiency characteristics and the same discount but with different reliabilities. Let us denote chatbots 1 and 2 and assume that both are sufficiently reliable in the sense that their overlook probabilities are quite small: $0 < a_i^s(s) < 0.5$, $s = 1, 2$, and in all locations, $i = 1, \ldots, N$. For definiteness, let us assume that the first chatbot is more reliable, that is, $0 < a_i^{(1)} < a_i^{(2)}$, $i = 1, \ldots, N$. Both chatbots operate in the same environment, but due to different reliabilities, their optimal search policies may be different.

Let $\boldsymbol{\pi}^{*(1)}$ be the optimal policy for chatbot 1 and $\boldsymbol{\pi}^{*(2)}$ be the optimal policy for chatbot 2. Accordingly, let $F_1(\boldsymbol{\pi}^{*(1)})$ and $F_2(\boldsymbol{\pi}^{*(2)})$ denote the satisfaction functions obtained by the first and second chatbot, respectively.

The main theoretical result of this work is as follows:

Proposition 2. *If the chatbots are sufficiently reliable, that is, $0 < a_i^{(1)} < a_i^{(2)}$, $i = 1, \ldots, N$, then, when using the optimal search policy, a less reliable chatbot is more efficient, in the sense that its search efficiency is greater than the efficiency of a more reliable chatbot, i.e., $F_1(\boldsymbol{\pi}^{*(1)}) < F_2(\boldsymbol{\pi}^{*(2)})$.*

To prove Proposition 2, we first need to prove the following auxiliary claim:

Proposition 3. *Let the function $f(x)$ be $x^{k-1}(1 - x)$, where k is a positive integer greater than 1. Then, $f(x)$ is increasing in x when $0 < x < 0.5$.*

Proof. We have: $f(x) = x^{k-1}(1 - x) = x^{k-1} - x^k$, $k > 1$.

Then, $f'(x) = (x^{k-1} - x^k)' = x^{k-2}(k - 1 - kx)$. Since $x > 0$, $f'(x) > 0$ iff $k - 1 - kx > 0$, or, equivalently, $f'(x) > 0$ iff $x < (k - 1)/k$.

Note that $(k - 1)/k \geq 0.5$ for any integer $k > 1$. Then, for $k > 1$, we have: $0 < x < 0.5 \leq (k - 1)/k$.

Therefore, $f'(x) > 0$, and, hence, $f(x)$ is increasing. □

Now, we return to the proof of Proposition 2. Let us denote the optimal expected discounted satisfactions for chatbots 1 and 2 by $F_1(\boldsymbol{\pi}^{*(1)}) = E(R_{disc,1}(\boldsymbol{\pi}^{*(1)}))$ and $F_2(\boldsymbol{\pi}^{*(2)}) = E(R_{disc,2}(\boldsymbol{\pi}^{*(2)}))$, respectively. Since the reliabilities of the two chatbots are different, the optimal search strategies of these smart robots, denoted as $\boldsymbol{\pi}^{*(1)} = (\boldsymbol{\pi}_1^{*(1)}, \boldsymbol{\pi}_2^{*(1)}, \ldots, \boldsymbol{\pi}_m^{*(1)}, \ldots)$ and $\boldsymbol{\pi}^{*(2)} = (\boldsymbol{\pi}_1^{*(2)}, \boldsymbol{\pi}_2^{*(2)}, \ldots, \boldsymbol{\pi}_m^{*(2)}, \ldots)$, respectively, in the general case can be different, despite all other characteristics of them being pairwise identical. The optimal values of the corresponding expected discounted satisfaction, $F_1(\boldsymbol{\pi}^{*(1)})$ and $F_2(\boldsymbol{\pi}^{*(2)})$, can also be different.

According to expression (1), we have:

$$F_1\left(\boldsymbol{\pi}^{*(1)}\right) - F_2\left(\boldsymbol{\pi}^{*(1)}\right) = \sum_{m=1}^{\infty} r_{\boldsymbol{\pi}_m^{*(1)}} e^{-d\sum_{k=1}^{m} t_{\boldsymbol{\pi}_k^{*(1)}}} \cdot \left(P_{\boldsymbol{\pi}_m^{*(1)}} p_{\boldsymbol{\pi}_m^{*(1)}} - P_{\boldsymbol{\pi}_m^{*(2)}} p_{\boldsymbol{\pi}_m^{*(1)}}\right) = \sum_{m=1}^{\infty} r_{\boldsymbol{\pi}_m^{*(1)}} e^{-d\sum_{k=1}^{m} t_{\boldsymbol{\pi}_k^{*(1)}}} \left[a_{\boldsymbol{\pi}_m^{*(1)}}^{u(\boldsymbol{\pi}^{*(1)},m)} \left(1 - a_{\boldsymbol{\pi}_m^{*(1)}}\right) p_{\boldsymbol{\pi}_m^{*(1)}} - a_{\boldsymbol{\pi}_m^{*(2)}}^{u(\boldsymbol{\pi}^{*(2)},m)} \left(1 - a_{\boldsymbol{\pi}_m^{*(2)}}\right) p_{\boldsymbol{\pi}_m^{*(1)}}\right]. \quad (5)$$

If $0 < a_{\boldsymbol{\pi}_m^{*(1)}} < a_{\boldsymbol{\pi}_m^{*(2)}} < 0.5$ for all m, then, according to Proposition 3, we have:

$$a_{\boldsymbol{\pi}_m^{*(1)}}^{u(\boldsymbol{\pi}^{*(1)},m)} \left(1 - a_{\boldsymbol{\pi}_m^{*(1)}}\right) p_{\boldsymbol{\pi}_m^{*(1)}} - a_{\boldsymbol{\pi}_m^{*(2)}}^{u(\boldsymbol{\pi}^{*(2)},m)} \left(1 - a_{\boldsymbol{\pi}_m^{*(2)}}\right) p_{\boldsymbol{\pi}_m^{*(1)}} \leq 0 \quad (6)$$

Then, from (5) and (6), it follows that if $0 < a_{\boldsymbol{\pi}_m^{*(1)}} < a_{\boldsymbol{\pi}_m^{*(2)}} < 0.5$ for all m, then the following is true:

$$F_1(\boldsymbol{\pi}^{*(1)}) \leq F_2(\boldsymbol{\pi}^{*(1)}). \quad (7)$$

Next, it is easy to see the following:

$$F_2(\boldsymbol{\pi}^{*(1)}) \leq F_2(\boldsymbol{\pi}^{*(2)}), \tag{8}$$

since $\boldsymbol{\pi}^{*(2)}$ denotes the optimal policy for chatbot 2, that is, $\boldsymbol{\pi}^{*(2)}$ guarantees the maximum expected satisfaction $F_2(\boldsymbol{\pi})$. From (7) and (8), it follows that a less reliable chatbot is more efficient in the sense that it provides greater expected satisfaction to the human user than a more reliable chatbot.

7. Discussion

At this point, two related questions arise: Does the paradoxical situation under consideration arise: (1) when the chatbot is completely reliable, i.e., its overlook probability $a_i = 0$, and (2) when the chatbot is not completely reliable, that is, its overlook probability is strictly positive and can be arbitrarily small?

Let us answer the first question first. A simple counterexample clearly shows that the answer here is: *there is no paradox*. For completeness, we provide proof below.

Consider the following numerical example, the idea of which was suggested to us by I.M. Sonin. For an absolutely reliable chatbot 1, we set $a_i{}^1 = 0$; for chatbot 2, we set $a_i{}^2 = 0.1$; the known satisfaction levels are $r_i = 10$ for each chatbot and any location, and the prior probabilities that the target information is hidden at location i are $p_i = 0.5$, $i = 1, 2$; $d = 0.1$. Then, the absolutely reliable chatbot 1 will inevitably find the target information at the right location, either during the first search with probability ½ or during the second search (after an unsuccessful first search) with probability ½, and the search process ends. Assuming $d = 0.1$, the overall objective function $F^1(\boldsymbol{\pi}^*)$ for the first chatbot is: $p_i \cdot 10 + p_i \cdot 10 \cdot e^{-0.1}$, that is, about 9.6.

As for the second, unreliable chatbot, since $a_i{}^2 > 0$, the iterative search in these two locations can have an infinite number of rounds, and the expected total satisfaction on an infinite horizon will obey expression (1). Elementary arithmetic operations on Formula (1) using the data for chatbot 2 and the sum of an infinite geometrical progression show that the expected satisfaction $F^2(\boldsymbol{\pi}^*)$ of the second chatbot does not exceed 4.9 points, and in fact, it will even be much less for a very small $a_i{}^2 > 0$. We see that the efficiency of an absolutely reliable chatbot is greater, which means *there is no paradox* in this case.

Now let us answer the second question. At first glance, and here too, the answer should not be paradoxical, since the objective function defined in expression (1) is continuous in the variable a_i, and, therefore, one would expect to obtain the same answer as in the case where the probability $a_i = 0$. However, this answer is incorrect—in fact, it is only true for the special case where the chatbot is absolutely reliable and retrieves the target information during the first round of the algorithm. In this case, the process is finite and terminates after one scan of each location. However, if the chatbot is unreliable and thus does not detect the target in the first round, then, starting from the second round of the multi-round iterative search, the objective function (1) is discontinuous in the discrete variable u_i (i.e., the number of unsuccessful scans of location i), and the proof based off the continuity of the objective function is no longer valid. The correct answer to the second question is that if the AI chatbot does not extract the target information in the first round of search, then during the iterative search procedure described in Section 5.2, the less reliable chatbot is more efficient—for any positive overlook probability a_i, even as small as you wish. This claim is rigorously proved in Proposition 2 of this section.

8. Numerical Examples

We conducted experimental computations in which we modeled the information extraction process for two AI chatbots with different characteristics; we varied the number of rounds of the algorithm, the scanning time, and the overlook probabilities, including

very large and very small ones. The computations consistently confirmed our judgement that the considered paradoxical phenomenon exists and is easily reproduced.

In this section, we considered numerical examples for two chatbots with six locations, randomly selected from our experiments. The examples convincingly confirmed the existence of the paradox. The input data are presented in Table 2. The notations are introduced in Section 3 as follows:

- p_i is a known probability that location i contains the information we are looking for; $i = 1, \ldots, 6$;
- t_i is the time (in minutes) spent by a human user in conversation with the chatbot to evaluate the chatbot's answer about the scan of location i; at each round, the time is assumed to remain the same;
- a_i is a non-zero overlook probability;
- r_i is a known Likert-scale satisfaction level obtained by the human user when the target information is found at location i (at any round) before discounting; recall that the level of satisfaction that a human user receives for a correct answer from the chatbot is measured on the 10-point Likert scale. This is the IBM SPSS rating system, in which the highest score of 10 signifies "very satisfied" on a 10-point scale ranging from 1 to 10, while the lowest score of 1 means "very dissatisfied" [4];
- $\boldsymbol{\pi} = (\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \ldots, \boldsymbol{\pi}_k)$ is a finite sequence of conversation stages (rounds);
- $\boldsymbol{\pi}_k$ is the number of a location (a database) scanned at the kth round of the sequence $\boldsymbol{\pi}$, $\boldsymbol{\pi}_k$ $A = \{1, \ldots, 6\}$, $k = 1, 2, \ldots$;
- n is the *a priori* unknown round of the sequence $\boldsymbol{\pi}$ at which the chatbot successfully extracts the targeted information;
- $T_{\boldsymbol{\pi},n} = \sum_{i=1}^{n} t_{\boldsymbol{\pi}_i}$ is the total time spent by a human user in conversation with the chatbot from round 1 to round n in sequence $\boldsymbol{\pi}$;
- If the search for the target information required $T_{\boldsymbol{\pi},n}$ time units in $\boldsymbol{\pi}$, the level of satisfaction in round $\boldsymbol{\pi}_n$ was discounted by $e^{-dT_{\boldsymbol{\pi},n}}$, for a known discount factor d, $d \in (0,1)$, i.e., the discounted level of satisfaction was computed as follows: $R_{discount}(\boldsymbol{\pi}) = r_{\boldsymbol{\pi}_n} e^{-dT_{\boldsymbol{\pi},n}} = r_{\boldsymbol{\pi}_n} e^{-d\sum_{i=1}^{n} t_{\boldsymbol{\pi}_i}}$.

Table 2. Input data, for example, $d = 0.1$.

Round \ Location	1	2	3	4	5	6
p_i	0.1	0.05	0.15	0.2	0.3	0.2
t_i (Min.)	3	2	4	1	3	5
r_i	5	10	4	7	2	10
$a_i^{(1)}$	0.2	0.05	0.2	0.04	0.1	0.3
$a_i^{(2)}$	0.3	0.1	0,4	0.1	0.2	0.35

The numerical parameters of the times and probabilities were randomly selected. These and numerous other examples stored in our dataset confirmed our theoretical judgements.

Calculations of the proposed iterative algorithm are presented in Tables 2 and 3 for chatbots 1 and 2, respectively, for 12 consecutive rounds. The table columns correspond to the six locations considered; the rows represent rounds of the search example considered. The elements in the cells are the coefficients $Q_{i,j}$, calculated for 12 rounds, row by row.

The number in the cell (i,j) of these tables represents the parameter $Q_{i,j}$, calculated by Formula (4), where $i = 1, \ldots, 12$; $j = 1, \ldots, 6$ for two chatbots. The tables represent the matrices M_k for two chatbots, introduced in Section 5.2. The calculations in Table 3 for

chatbot 1 can be observed; similar calculations were performed in Table 4 for chatbot 2. For chatbot 1, in the first round, the first row of matrix M_1 (i.e., the first row of Table 3) was filled in with the coefficients $Q_{i,1}$, $i = 1, \ldots, N$. Then, all the locations were (unsuccessfully) scanned once, after which the $Q_{i^*(1),1} = max_i Q_{i,1} = Q_{4,1} = 0.5112$ and the corresponding location $i^*(1) = 4$ were found. Next, during this round, chatbot 1 scanned location 4 once more; we assumed that in this example, it did not discover the relevant target information there. Then, the number of unsuccessful scans of this location increased to 2, and the corresponding parameter $Q_{4,2}$ was calculated according to (4), (equals 0.0204) and placed in the cell (2,4) of matrix M_2. All other elements of the second row of this matrix were left unchanged—the same as in the previous row of M_1. The first round of the algorithm was completed, and then the second round started.

Table 3. Calculations of the optimal policy for chatbot 1.

Round \ Location	1	2	3	4	5	6	π^1	$F_1(\pi_1)$
1	0.2287	0.1073	0.1952	**0.5112**	0.1543	0.3237	4	0.0486
2	0.2287	0.1073	0.1952	0.0204	0.1543	**0.3237**	6	0.1639
3	**0.2287**	0.1073	0.1952	0.0204	0.1543	0.0971	1	0.1964
4	0.0457	0.1073	**0.1952**	0.0204	0.1543	0.0971	3	0.2226
5	0.0457	0.1073	0.0390	0.0204	**0.1543**	0.0971	5	0.2237
6	0.0457	**0.1073**	0.0390	0.0204	0.0154	0.0971	2	0.2239
7	0.0457	0.0054	0.0390	0.0204	0.0154	**0.0971**	6	0.2258
8	**0.0457**	0.0003	0.0390	0.0204	0.0154	0.0291	1	0.2270
9	0.0091	0.0003	**0.0390**	0.0204	0.0154	0.0291	3	0.227004
10	0.0091	0.0003	0.0104	**0.0204**	0.0154	0.0291	4	0.227008
11	0.0091	0.0003	0.0104	0.0008	0.0154	**0.0291**	6	0.227312
12	0.0091	0.0003	0.0104	0.0008	**0.0154**	0.0087	4	0.227321

Table 4. Calculations of the optimal policy for chatbot 2.

Round \ Location	1	2	3	4	5	6	π^2	$F_2(\pi_2)$
1	0.3001	0.2032	0.2928	**1.1980**	0.2744	0.3507	4	0.1140
2	0.3001	0.2032	0.2928	0.1198	0.2744	**0.3507**	6	0.2389
3	**0.3001**	0.2032	0.2928	0.1198	0.2744	0.1227	1	0.2816
4	0.0900	0.2032	**0.2928**	0.1198	0.2744	0.1227	3	0.3208
5	0.0900	0.2032	0.1171	0.1198	**0.2744**	0.1227	5	0.3402
6	0.0900	**0.2032**	0.1171	0.1198	0.0549	0.1227	2	0.3476
7	0.0900	0.0203	0.1171	0.1198	0.0549	**0.1227**	6	0.3562
8	0.0900	0.0203	0.1171	**0.1198**	0.0549	0.0430	4	0.3574
9	0.0900	0.0203	**0.1171**	0.0120	0.0549	0.0430	3	0.3591
10	0.0900	0.0203	0.0187	0.0120	**0.0549**	0.0430	5	0.359311
11	**0.0900**	0.0203	0.0187	0.0120	0.0110	0.0430	1	0.359696
12	0.0027	0.0203	0.0187	0.0120	0.0110	**0.0430**	6	0.360275

In round 2, the next location to be scanned was selected by computing $Q_{i^*(2),2} = \max_i Q_{i,2} = Q_{6,2} = 0.3237$, and hence, the location $i^*(2) = 6$ was chosen. $Q_{i^*(2),2}$ was placed in cell (2,6) of the second row of matrix M_2, and this item was highlighted in bold.

Next, during this round, chatbot 1 scanned location 6 once more; we assumed that the human user was not satisfied by the target information extracted there. Then, the number of unsuccessful scans of this location increased to 2; the corresponding parameter $Q_{6,3}$ was calculated according to (4) (equals 0.0204) and placed in cell (3,6) of the next matrix M_3. All other elements of the third row of this matrix were left unchanged—the same as in the previous row of M_2. The second round was completed.

During the subsequent rounds, the algorithm sequentially filled in the next rows 3, 4, . . ., 12 of Table 3, following the instructions of the algorithm, as in round 2.

We assumed that the human user was satisfied by the target information retrieved in the 12th round. The results obtained for each chatbot over 12 rounds were as follows: (1) The optimal policies for chatbots 1 and 2 were equal to $\boldsymbol{\pi}^{*(1)}$ = (4, 6, 1, 3, 5, 2, 6, 1, 3, 4, 6, 5) and $\boldsymbol{\pi}^{*(2)}$ = (4, 6, 1, 3, 5, 2, 6, 4, 3, 5, 1, 6), respectively. (2) The partial efficiency $F_1(\boldsymbol{\pi}^{*(1)})$ and, accordingly, $F_2(\boldsymbol{\pi}^{*(2)})$ obtained at each round for the two chatbots are shown in the corresponding cells of the last columns of Tables 3 and 4, respectively. Comparing these results, we see that *in each round* of the search process, even in the early rounds, the optimal expected search efficiency for a less reliable chatbot was significantly greater than the optimum efficiency of the more reliable chatbot. The example considered, constructed for 12 rounds, covered simpler numerical examples; for example, the calculations presented in the top rows 1, 2, and 3 of Tables 3 and 4 corresponded to a small illustrative example where the number of rounds of the algorithm was 3.

These numerical examples and, more generally, all the other numerous experimental calculations that we systematically performed showed that the paradoxical phenomenon in question clearly exists and is easily reproduced. However, it is unknown whether the experimental sensor data always reflect the possible behavior of real, accessible AI chatbots. Investigating this issue is a challenging open question and an attractive direction for future research.

9. Concluding Remarks and Open Problems

The focus of this work was to study a realistic situation where the overlook probability of a chatbot is non-zero and thus, the search process does not end during the first round of human–chatbot conversation. The main contributions of this paper were as follows. First, we analyzed the two-player discrete sequential search problem under study, derived a greedy index-based iterative search policy for its solution, and proved that it was optimal for the considered search problem. Then, we identified a paradoxical "more-for-less" phenomenon: a less reliable chatbot had a higher efficiency for any positive overlook probability, even if it was very small.

Despite the "paradoxical" wording of the title, the result of this paper is quite logically explainable and completely compatible with common sense. The point is that if a more reliable chatbot has a lower overlook probability, then it is plausible to think that it will find the target information faster, i.e., in fewer search rounds and ultimately in less average search time.

This common-sense observation is entirely consistent with the well-known fact from probability theory about the random variable x, which is called the geometric random variable with parameter p and which represents the random number of independent unsuccessful trials (rounds) before the target is found; this probability is given by $p(k) = p(1-p)^{k-1}$, $k = 1, 2, \ldots$ In this case, it is known that the expected number of trials equals $1/\mathrm{p} = 1/(1 - a_i)$ [34]. Thus, the higher the overlook probability, the greater the

expected number of rounds and the average search time before the target information is found, and, therefore, the greater the total expected accumulated information and the search efficiency determined by expression (1). Therefore, in essence, there is no contradiction in the obtained result.

Finally, we would like to acknowledge some limitations of the present study, most notably the lack of comparison between the proposed mathematical model of information extraction with AI and the behavior of real, publicly available AI chatbots such as ChatGPT; the lack of discussion of potential bias in human users' satisfaction assessment; and the lack of information about chatbots using non-greedy search policies where data are not scanned sequentially. These issues are beyond the scope of this study and are challenging directions for future research.

It would be interesting to know what paradoxical situations may arise for other search/extraction scenarios that arise in information search performed by real AI chatbots, in particular for multi-target, multi-criteria, multi-user, and multi-chatbot scenarios. Another general problem is to find and resolve real-world situations for good search policies (ideally, optimal or approximate with performance guarantees) based on advanced algorithmic ideas such as optimizing with learning [35] and reinforcement search [36].

Author Contributions: Both authors contributed to the study conception and design. Material preparation, data collection, and analysis were performed by B.K. and E.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Acknowledgments: The authors would like to thank I.M. Sonin for helpful discussions and the assistant editors and four anonymous reviewers for very useful comments and suggestions.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Nicolescu, L.; Tudorache, M.T. Human-Computer Interaction in Customer Service: The Experience with AI Chatbots—A Systematic Literature Review. *Electronics* **2022**, *11*, 1579. [CrossRef]
2. Izadi, S.; Forouzanfar, M. Error Correction and Adaptation in Conversational AI: A Review of Techniques and Applications in Chatbots. *AI* **2024**, *5*, 803–841. [CrossRef]
3. Akdemir, D.M.; Bulut, Z.A. Business and Customer-Based Chatbot Activities: The Role of Customer Satisfaction in Online Purchase Intention and Intention to Reuse Chatbots. *J. Theor. Appl. Electron. Commer. Res.* **2024**, *19*, 2961–2979. [CrossRef]
4. Alabbas, A.; Alomar, K. A Weighted Composite Metric for Evaluating User Experience in Educational Chatbots: Balancing Usability, Engagement, and Effectiveness. *Future Internet* **2025**, *17*, 64. [CrossRef]
5. Ross, S.M. A problem in optimal search and stop. *Oper. Res.* **1969**, *17*, 984–992. [CrossRef]
6. Sweat, C.W. Sequential Search with Discounted Income, the Discount a Function of the Cell Searched. *Ann. Math. Stat.* **1970**, *41*, 1446–1455. [CrossRef]
7. Chew, M.C. Optimal Stopping in a Discrete Search Problem. *Oper. Res.* **1973**, *21*, 741–747. [CrossRef]
8. Ross, S.M. *Introduction to Stochastic Dynamic Programming*; Academic Press: New York, NY, USA, 1983; pp. 63–71.
9. Levner, E. Infinite-horizon scheduling algorithms for optimal search for hidden objects. *Int. Trans. Oper. Res.* **1994**, *1*, 241–250.
10. Cheng, T.C.E.; Kriheli, B.; Levner, E.; Ng, C.T. Scheduling an autonomous robot searching for hidden targets. In *Annals of Operations Research*; Springer: Berlin/Heidelberg, Germany, 2019; Volume 298, pp. 95–109.
11. Koopman, B. The theory of search, II. Target detection. *Oper. Res.* **1956**, *4*, 324–346. [CrossRef]
12. Stone, L.D. *Theory of Optimal Search*; Elsavier: Amsterdam, The Netherlands, 1976; Volume 71, p. 1008.
13. Stone, L.D.; Royset, J.O.; Washburn, A.R. *Optimal Search for Moving Targets*; Springer Nature: Cham, The Netherlands, 2016.
14. Trummel, K.E.; Weisinger, J.R. Technical Note—The Complexity of the Optimal Searcher Path Problem. *Oper. Res.* **1986**, *34*, 324–327. [CrossRef]
15. Wegener, I. Optimal search with positive switch cost is NP-hard. *Inf. Process. Lett.* **1985**, *21*, 49–52. [CrossRef]

16. Kriheli, B.; Levner, E.; Spivak, A. Optimal Search for Hidden Targets by Unmanned Aerial Vehicles under Imperfect Inspections. *Am. J. Oper. Res.* **2016**, *6*, 153–166. [CrossRef]
17. Kress, M.; Lin, K.Y.; Szechtman, R. Optimal discrete search with imperfect specificity. *Math. Methods Oper. Res.* **2008**, *68*, 539–549. [CrossRef]
18. Osborne, M.J.; Rubinstein, A. *A Course in Game Theory*; MIT Press: Cambridge, MA, USA, 1994.
19. Clarkson, J.; Lin, K.Y.; Glazebrook, K.D. A classical search game in discrete locations. *arXiv* **2021**, arXiv:2103.09310. [CrossRef]
20. Yang, C.; Zhu, Y.; Chen, Y. A Review of Human–Machine Cooperation in the Robotics Domain. *IEEE Trans. Human-Machine Syst.* **2021**, *52*, 12–25. [CrossRef]
21. Bansal, S.; Xu, J.; Howard, A.; Isbell, C. Bayes–Nash: Bayesian inference for Nash equilibrium selection in human-robot parallel play. *Auton. Robot.* **2021**, *46*, 217–230. [CrossRef]
22. Tseng, K.-S.; Mettler, B. Analysis and augmentation of human performance on telerobotic search problems. *IEEE Access* **2020**, *8*, 56590–56606. [CrossRef]
23. Robinson, N.; Williams, J.; Howard, D.; Tidd, B.; Talbot, F.; Wood, B.; Pitt, A.; Kottege, N.; Kulić, D. Human-Robot Team Performance Compared to Full Robot Autonomy in 16 Real-World Search and Rescue Missions: Adaptation of the DARPA Subterranean Challenge. *ACM Trans. Human-Robot Interact.* **2024**, *14*, 1–30. [CrossRef]
24. Steinberg, R.; Zangwill, W.I. The Prevalence of Braess' Paradox. *Transp. Sci.* **1983**, *17*, 301–318. [CrossRef]
25. Shan, F.; Li, H.; Wang, Z.; Jin, M.; Chen, D. Optimizing Rural Highway Maintenance Scheme with Mathematical Programming. *Appl. Sci.* **2024**, *14*, 8253. [CrossRef]
26. Schäfer, B.; Pesch, T.; Manik, D.; Gollenstede, J.; Lin, G.; Beck, H.-P.; Witthaut, D.; Timme, M. Understanding Braess' paradox in power grids. *Nat. Commun.* **2022**, *13*, 5396. [CrossRef]
27. Gershenson, C.; Helbing, D. When slower is faster. *Complexity* **2015**, *21*, 9–15. [CrossRef]
28. Boaz, T.; Ben-Porat, O. Braess's Paradox of Generative AI. *arXiv* **2024**, arXiv:2409.05506.
29. Sorrell, S. *Exploring Jevons' Paradox. Energy Efficiency and Sustainable Consumption: The Rebound Effect*; Palgrave Macmillan: London, UK, 2009; pp. 136–164.
30. York, R.; McGee, J.A. Understanding the Jevons paradox. *Environ. Sociol.* **2015**, *2*, 77–87. [CrossRef]
31. Spieksma, F.C.; Woeginger, G.J. The no-wait flow-shop paradox. *Oper. Res. Lett.* **2005**, *33*, 603–608. [CrossRef]
32. Charnes, A.; Klingman, D. The more-for-less paradox in the distribution model. *Cah. Cent. D'etudes Re-Cherch. Opérationnelle* **1971**, *13*, 11–22.
33. Graham, R.L. Bounds on Multiprocessing Timing Anomalies. *SIAM J. Appl. Math.* **1969**, *17*, 416–429. [CrossRef]
34. Ross, S.M. *A First Course in Probability*; Macmillan: New York, NY, USA, 1976.
35. Powell, W.B. Perspectives of approximate dynamic programming. In *Annals of Operations Research*; Springer: Berlin/Heidelberg, Germany, 2012; Volume 298, pp. 319–356.
36. Zhang, T.; Mo, H. Reinforcement learning for robot research: A comprehensive review and open issues. *Int. J. Adv. Robot. Syst.* **2021**, *18*, 17298814211007305. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.