Natural Language Meets Database: System Transparency and User Understanding in NL-to-SQL Translation

Bergische Universität Wuppertal

THIVYAN SIVANANTHAN, JACOB ORTENBERG, and PARSSA JASHNIEH

Typically, when a user interacts with a website, such as searching for an item or filtering the search results, a SQL template is used for which the user input gets added. By using an AI that translates natural language into SQL queries, the initial search results based on the keywords passed will most likely be tailored to the user's wishes, providing better search results initially without the need to interact with filters. To ensure regulatory compliance, such as not being able to use passwords as a search criterion, we assert a privacy preserving AI that can easily be integrated into an existing code/database.

CCS Concepts: • **Human-centered computing** → *Web-based interaction*; Interaction design theory, concepts and paradigms.

Additional Key Words and Phrases: AI, Seq2Seq, SQL, UI, UX, Filters, Search, Transfer Learning, NLP

ACM Reference Format:
Thivyan Sivananthan, Jacob Ortenberg, and Parssa Jashnieh. 2025. Natural Language Meets Database: System Transparency and User Understanding in NL-to-SQL Translation. In *Final Project for the HIC Module at the Bergische Universität Wuppertal(Human-Computer Interaction & Artificial Intelligence), October 7–February 28, 2025, Wuppertal, Germany*. ACM, Wuppertal, North Rhine Westphalia, Germany, 11 pages.

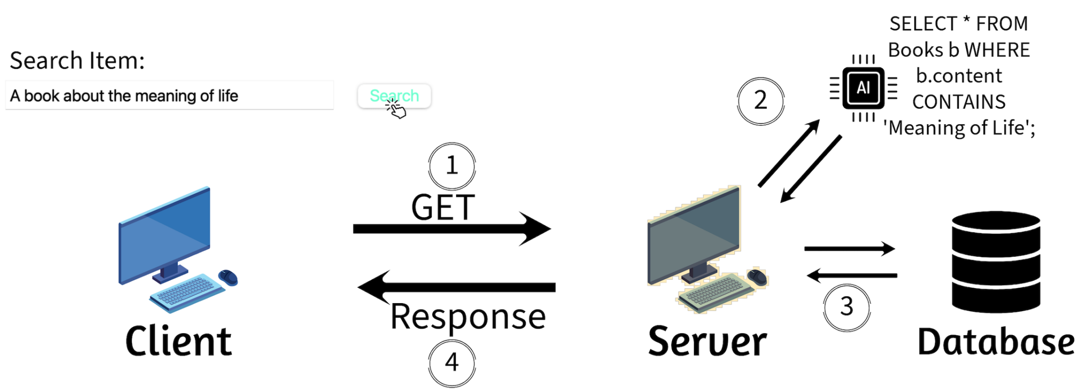

Fig. 1. Highlighting the interaction paradigm of the user with an AI. Instead of traditionally passing the users input to a static SQL-Template, it is passed to an AI-System that interpretes the users search request and dynamically generates a SQL-Query . The resulting query is then executed and displayed for the user in the Interface.

Authors' Contact Information: Thivyan Sivananthan; Jacob Ortenberg; Parssa JashniehBergische Universität Wuppertal.

Manuscript submitted to ACM

1 Introduction

Online shopping platforms like Amazon and eBay have become the go-to sources for a vast range of products. However, users often face difficulties when searching for specific items, especially products with more complex or technical specifications, such as electronic devices, computer components, or specialized tools. These challenges can lead to frustration and even cause users to switch platforms in search of a better experience. The root of these difficulties lies in the individuality of search behavior, which varies between users. Some rely on generic search terms, such as "laptop," while others prefer highly specific queries, like "Dell XPS 15 9530 with Intel Core i7 and 32GB RAM." Our study found that search behavior is roughly evenly divided between these two approaches, making it essential for e-commerce platforms to optimize their search functionalities to accommodate diverse user needs. At the same time, this presents a challenge, as creating a platform that satisfies all users equally is difficult. In addition to search queries, filtering options play a crucial role in the online shopping experience. Filters help users refine their results based on criteria such as price, size, brand, technical specifications, or customer ratings. However, ineffective or overly complex filtering systems can hinder rather than enhance the search process. If filters do not align with user expectations, are difficult to use, or fail to narrow down results efficiently, users may become frustrated and abandon their search. Therefore, improving search functionalities is essential to enhance user experience, increase customer satisfaction and ensure platform loyalty. Building upon these challenges, we propose an alternative approach to traditional search mechanisms. Instead of relying on predefined SQL templates, we suggest dynamically generating SQL queries using machine learning techniques. By leveraging AI to generate queries in real time, the system can automatically interpret the user's intent based on their input, providing more precise and relevant search results initially. If the AI performs at a high level, this approach could eventually eliminate the need for manual filtering mechanisms, as the AI would inherently incorporate these preferences into the generated query. This shift could significantly enhance human-computer interaction, making the search process more intuitive, efficient, and user-centric. To determine whether such a scenario is truly possible and to understand how users would interact with this approach, we conducted a study. To carry out this study, we implemented a prototype interface to evaluate user behavior and gather insights on the effectiveness of dynamically generated queries in direct comparison to traditional static templates.

1.1 The conventional search mechanism

The Open Worldwide Application Security Project (OWASP) regularly updates its top 10 list of web application security risks. SQL Injection attacks have consistently remained in this list across multiple iterations, including the 2017 and 2021 versions, largely due to the widespread use of custom SQL query templates in web applications[2]. These templates follow predefined schemas where user input, such as search queries, is inserted into predetermined placeholders. This approach presents multiple significant limitations. These templates are vulnerable to SQL Injection attacks (SQLIA) if not properly secured. Additionally, their search capabilities are inherently restricted, as the searchable attributes or columns must be explicitly defined by developers based on specific requirements. While these templates excel at simple keyword matching, they struggle with natural language queries due to their rigid, predefined structure. Although this keyword-based approach can efficiently process basic search terms, users typically need to rely on additional filtering mechanisms to narrow down their results to find exactly what they are looking for. However, these filtering systems come with their own set of limitations.

1.2 Limitations of traditional filtering mechanisms

Traditional e-commerce platforms typically implement filtering systems through a combination of predefined dropdown menus, checkboxes, and range selectors. While these mechanisms provide basic functionality, they present several significant challenges. First, the static nature of these filters requires regular maintenance and updates to accommodate new product categories and attributes. Second, the presentation of filter options often follows a one-size-fits-all approach, disregarding individual user preferences and shopping patterns. For instance, technical users searching for computer components might prefer detailed specification filters, while casual shoppers might find such granularity overwhelming. Furthermore, traditional filtering systems often struggle with semantic relationships between different product attributes. A user searching for a "gaming laptop" might need to separately filter for high RAM, dedicated graphics card, and fast processor, even though these specifications are inherently related to the concept of gaming performance. This disconnect between user intent and filtering capabilities becomes particularly problematic when dealing with complex product categories that have numerous interdependent specifications, such as electronic devices or specialized equipment.

1.3 Moving beyond traditional search limitations

These limitations of conventional search templates and filtering mechanisms highlight the need for more sophisticated approaches that can adapt to diverse user needs. Modern developments in database querying, machine learning, and natural language processing offer promising directions for addressing these challenges through more flexible and intelligent search solutions.
Based on the identified challenges in e-commerce search mechanisms, we formulate three main research questions that guide our investigation.

(1) We examine how natural language interfaces affect user search behavior and satisfaction compared to traditional keyword-based approaches.
(2) We investigate the technical feasibility of translating diverse natural language queries into SQL statements.
(3) We analyze user preferences and requirements regarding correction mechanisms in AI-driven search systems. This investigation focuses on understanding which types of refinement options users would find most helpful when initial search results don't meet their expectations, and how these correction possibilities influence their trust in the system.

2 Background

Translating natural Text into SQL-Queries is not a new undertaking. Back in 2003 Ana-MariaPopescu et.al proposed the PRECISE-System, which translates text into SQL-Queries based on a derived rule system. Not only did the PRECISE-System achieve remarkable Results, but the authors also mention that the translation of natural text into SQL-Queries is considered an AI-Complete task. With the rapid advancements in AI in recent years, it makes sense to replace the deterministic rule based system with a foundation model like CHAT-GPT to generate the SQL-Queries instead.

The objective of this initiative is to enhance the effectiveness of the search function on websites, with a particular focus on enhancing the performance of the input box. As outlined in [3] , 69% of users initiate a search for products directly via the search bar. However, 80% of these users subsequently depart from the webpage due to unsatisfactory experiences with the search functionality. In this study, 41% of respondents expressed frustration regarding the display of irrelevant content. A further issue that was identified as a point of frustration for 32% of users was the presentation of products that were no longer available. A further issue that 26% of users encounter with contemporary search systems on websites is the inability of these systems to accurately interpret user queries. To illustrate, when one seeks to procure a dresser and inaccurately enters "closet" as the search term, the websites encounter difficulties in locating the desired item.

A total of 69% of users have confirmed that they encounter irrelevant search results, and of those, 35% have indicated that this behavior prompts them to leave the webpage. A further 27% of users have confirmed that they have abandoned webpages due to the inability to restrict their searches. An additional 26% have done so due to out-of-stock status. A further notable issue is the absence of error tolerance in current websites, which results in the failure to display outcomes in instances of typographical inaccuracy.

2.1 Translation difficulties between text and SQL(A)

Achieving an accurate interpretation of the user's intent in the context of translating text into SQL poses a considerable challenge. This is due to the fact that some words could be ambiguous or the user might use synonyms. The system should also demonstrate fault-tolerance capabilities, as it is expected to generate a query in the event of typing errors. As previously stated in [1], it is imperative to exercise caution to ensure that the system does not generate erroneous queries. Otherwise, there is a risk of diminishing user trust in the website. As elaborated upon in [1], users demonstrate a reluctance to trade a user interface that is characterised by reliability and predictability for an intelligent system that is deemed unreliable.

2.2 Privacy concerns(A)

Furthermore, it must be considered that non-technical users can also exploit the system by requesting private data. This scenario poses significant privacy concerns, given that the users can bypass any need to understand SQL, and instead simply request the data they desire from the LLM. In such systems, the implementation of measures designed to restrict access to data is essential for ensuring the security of private data. According to the Open Worldwide Application Security Project (OWASP), which is a non-profit organization that aims to improve the security of software projects, SQL-Injections are still in the Top 10 Security Risks of 2021 and it is likely that they will remain in the Top 10 of 2025, taking into account that they were the number one attack in the Top 10 of 2017 [2].

2.3 Problem with existing text-to-sql(A)

The rule-based method for generating SQL queries was presented in the paper [1]. However, several issues were identified in the course of this approach, which were addressed to a limited extent. As previously mentioned, the system is confronted with ambiguous words. In [1], the recommendation was made to present the user with all possible variants that can be derived from the multi-word term. This enables the user to make an informed decision, ensuring an appropriate interpretation of their intention. A further challenge arises when a word is not present in the lexicon, as no query is generated in this instance. In such a scenario, it is imperative for the user to reformulate their query in a manner that is comprehensible to the system. However, this approach introduces a significant disadvantage, as the preliminary query may be processed by the system, but its execution is precluded due to the system's limited capabilities. The maintenance of synonyms poses a considerable challenge, as it is necessary to provide potential users with the opportunity to consult them. Consequently, a continuous adaptation of the system is required to ensure the quality of the results of the searches. As demonstrated in [1], a further issue is the considerable duration of user inquiry processing. In this study, a processing duration of six seconds was observed, which is regarded as relatively extensive when compared to the duration of a search in a web-based store. Nonetheless, a pivotal element was absent from the previously mentioned papers: the context of the values stored in the database. To illustrate, supplementary information becomes pertinent when storing the age of dogs. This necessity arises from the potential for storing the age of dogs in either months or years. Depending on the specific context, additional rules may be in place that must be taken into consideration.

Text-to-SQL systems that leverage LLMs have been developed; a notable example is SQLAI.AI [1]. Despite its development for developers, SQLAI.AI is not suitable for practical application.

3 Methods

For our study, we implemented a natural language interface in the form of a webshop that allows users to directly compare our approach of dynamically generated queries with a traditional static template. This setup lets users try both methods directly and compare how well they retrieve relevant search results. For our example webshop, we chose to focus on a single category of items to ensure a more controlled and meaningful analysis. We ultimately decided on a webshop offering dogs, as dogs are highly popular and provide a diverse range of characteristics for comparison. The interface was structured so that the displayed search results were divided into two distinct sections. On the left side, the results were generated using AI-driven, dynamically created search queries, while on the right side, the results were retrieved using a traditional, static query template, incorporating a semantic matching function, as observed on various websites. The user entered their search query through a shared search bar. We decided to apply the filtering functionality to both sections via a unified sidebar to avoid user confusion, even though it was not strictly necessary for the AI-driven side, as the constraints covered by the filters could also be expressed in natural language. This design enabled users to directly compare both approaches in real time. By reviewing both result sets side by side, participants could evaluate the effectiveness of each method and determine which approach best aligned with their personal search behavior and preferences.

3.1 Implementation of the Interface

Our initial approach for dynamically generating SQL queries was to use a local large language model. To further refine the performance, we intended to implement the DAIL schema. However, the main issue with using a local model was that it had to be loaded into the computer's RAM, which limited us to smaller models due to hardware constraints. These smaller models, however, did not deliver the expected performance in terms of both the accuracy of the generated SQL queries and the time required to generate them. As a result, we explored the possibility of using an API-based solution. For this, we tested both the Claude API and the OpenAI API. After extensive testing, we found that the Claude API performed slightly better. Consequently, we decided to integrate the Claude API into our system. The API delivered such strong results for our interface that, due to time constraints and the additional complexity involved, we ultimately decided not to implement the DAIL schema. In the prompt we used to dynamically generate SQL queries, we provided the API with both the database schema and all possible attribute values. This allowed the AI to correctly interpret user input and generate the appropriate query, even when the input was vague or descriptive. Additionally, we included a "base query" in the prompt, which outlined a general structure for the SQL query that the AI would then refine and complete based on the user's input.

[1]https://www.sqlai.ai/

3.2 User Study

In the course of our user study, a group of 13 participants was examined, consisting of 12 males and one female. The majority of the participants were currently enrolled in a Bachelor's or Master's program in Computer Science. Additionally, most had prior experience with artificial intelligence, either through academic studies or personal interest. Before the study began, all participants were informed about its primary objective, which was to evaluate the performance of the AI-driven search system in direct comparison to a conventional, template-based search mechanism. By conducting this study, we aimed to determine whether the AI system could produce superior results in product search. The participants were asked to answer a predefined set of questions (see Appendix) to assess their experience and the system's performance. The user study was divided into three phases. In the first phase, participants were asked a series of general questions before interacting with the interface. These questions aimed to gather insights into their past search behavior, including any difficulties or challenges they had encountered when searching for specific items and the consequences these issues had on their overall shopping experience. In the second phase, participants were introduced to our interface and given the opportunity to explore its functionality. To familiarize themselves with the system, participants were first asked to complete a set of predefined search tasks. These tasks aimed to provide an understanding of the interface's functionality and the variety of dogs and their attributes. They were allowed to conduct these searches freely, choosing between traditional static filters and natural language queries. The tasks were designed so that the first few could be solved relatively easily using filter settings, while the later ones were either very difficult or even impossible to accomplish with filters alone—for example, "OR" queries. This ensured that participants would engage with the AI-based system and recognize its unique and superior capabilities compared to the static template. Participants also had the option to refine or correct their queries if the initial results were unsatisfactory. After this guided introduction, they were encouraged to experiment with their own complex search queries using natural language. The final phase of the study involved a more detailed questionnaire designed to evaluate the AI-driven approach in comparison to the conventional model. To assess participant satisfaction with our interface and the AI-based approach, we included several questions in this part of the study, which were rated on a 5-point Likert scale. We also investigated participants' perspectives on the role of filtering mechanisms, exploring whether a model like ours could serve as a potential replacement for traditional filtering functionalities.

4 Result

This section presents the findings of our user study, evaluating the performance of an AI-driven natural language interface (NLI) against a conventional search system in an e-commerce context. The results address user satisfaction, the role of filters, technical feasibility, and possible correction mechanisms for query optimization, providing insights into the effectiveness of dynamic SQL query generation compared to static templates.

4.1 User Satisfaction and Search Behavior(RSQ1)

As shown in Figure 2, all participants expressed higher satisfaction with the AI-driven system's results, with responses on a 5-point Likert scale ranging from 'satisfied' to 'very satisfied.' Specifically, 69% (9 out of 13 participants) rated their satisfaction as very high (1/5), and 31% (4 out of 13) as high (2/5). The mean satisfaction score was 1.69 (SD = 0.48, calculated with bessel correction), indicating low variability in responses. Due to the small sample size (n=13), statistical significance tests were not conducted. Users highlighted the system's ability to accurately interpret their search intent as a key factor in their positive experience. Notably, the AI's capability to handle typographical errors and synonyms was well-received. For example, it recognized 'Wiener Dog' as 'Dachshund,' as shown in Figure 4. This feature was absent in the conventional system.

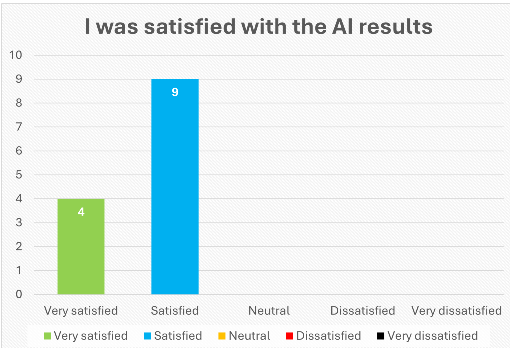

Fig. 2. The users' rating regarding the satisfaction of the results displayed by the AI. As can be seen in the picture, the participants were very satisfied with the results of the AI.

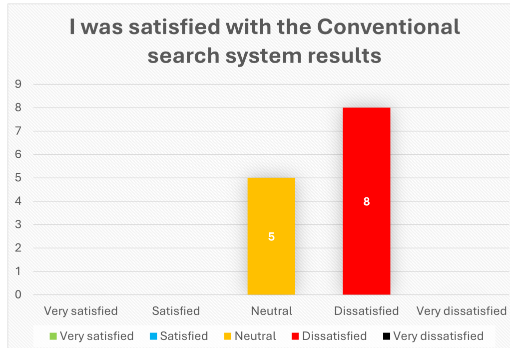

Fig. 3. The evaluation of the users regarding the satisfaction with the result displayed by the conventional system. The picture shows that they were not satisfied with the results displayed by the conventional system.

In contrast, satisfaction with the conventional system was markedly lower, averaging 3.62(SD = 0.51) on the same scale, with only 38% (5 out of 13) rating it as neutral (3/5), and 62% (7 out of 13) as dissatisfied (4/5), with no participants rating it as very high,high or very dissatisfied (see Figure 3). Users found its performance lacking, particularly when compared to the AI's intuitive query handling. An interesting observation emerged: as participants grew accustomed to the AI's ability to understand their intent, their reliance on manual filters decreased. This shift likely contributed to the conventional system's perceived inferiority, as it depended heavily on filter usage to refine results. This rapid adaptation suggests a potential change in search behavior, though further analysis is needed to confirm its extent.

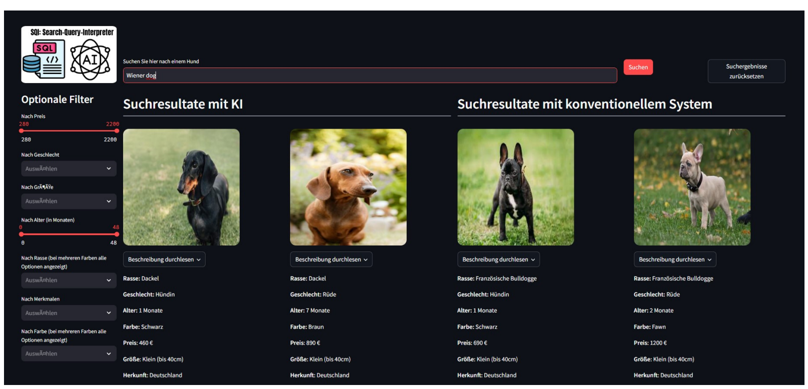

Fig. 4. In this picture we see the website that was developed for the project. The website was divided so that the user can quickly see which results belong to AI and which to the conventional system. To demonstrate the performance of the AI, Wiener dog was entered in the input field. As you can see, the AI can handle this, but the conventional system cannot.

It's worth noting that the results for the conventional system should be considered in light of the relatively simple static template we used. This basic design may have contributed to its poorer performance, and a more sophisticated static template could potentially have yielded better outcomes.

4.2 The Role of Filters in Enhancing User Experience

Despite the AI's potential to replace traditional filtering mechanisms, our study revealed that filters retained significant relevance for users. Figure 5 shows participants' preferences regarding filter usage in the AI-driven system, based on a 5-point Likert scale (1 = ‚strongly agree', 5 = ‚strongly disagree') from the post-study questionnaire. Of the 13 participants, 9 (69.2%) agreed or strongly agreed they could manage without filters (scores 1–2), 3 (23.1%) disagreed, preferring to retain them (scores 4), and 1 (7.7%) selected ‚neutral' (score 3). Of the participants who preferred filters, three stated in open-ended comments that they appreciated the visual clarity, such as when sorting by price or selecting ranges (e.g., 'price between 50-100€'), compared to text-based inputs. For instance, participants noted that setting a price range via a slider was faster and more intuitive than specifying it textually. These findings, however, are based on a small, homogeneous sample of 13 participants, primarily computer science students, which may limit generalizability.

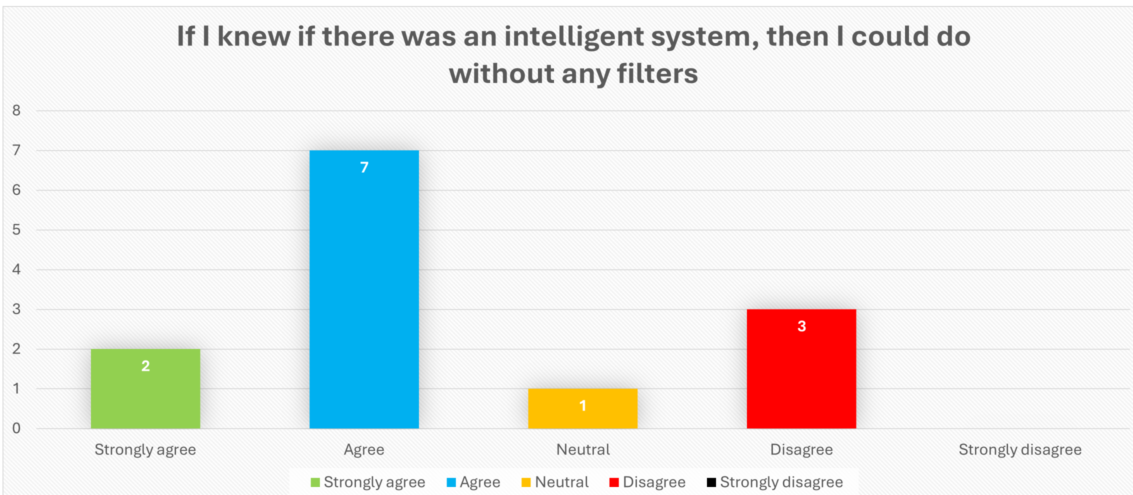

Fig. 5. After the users have used the system, they should state whether they would manage without the filter.The results show that the majority of users can manage without a filter without any problems. Three times "disagree" was indicated, while no "strongly disagree" was recorded.

4.3 Technical Feasibility and Security Considerations

Building on user experience insights, this subsection examines the technical implications of the AI-driven system. During the study, some participants tested the system's boundaries, revealing critical security concerns. For example, one user entered a command to 'delete all tables,' which the AI translated into an executable SQL query—an unintended capability that exposed a vulnerability. Ideally, the system should restrict operations to read-only searches, preventing modifications to the database or its schema. While our simple dog webshop schema excluded sensitive data like user information, real-world applications would demand robust safeguards. We propose implementing a security mechanism using Prepared Statements, which bind user inputs to predefined query structures, preventing malicious commands from being executed. Alternatively, a whitelist of permitted SQL operations (e.g., SELECT only) could ensure that generated queries remain safe, enhancing both functionality and security.

To address this, we propose implementing a log file-based blacklist to enhance security. This approach would involve maintaining a record of prohibited words, such as specific table names or keywords like "delete" or "drop," that the AI is not permitted to include in its output. Before executing any generated SQL query, the AI could cross-check its commands against this blacklist. If a match is detected, the query would be blocked, preventing potentially harmful operations. This mechanism would safeguard both the integrity of the data and the security of the database, offering a practical layer of protection while preserving the system's functionality for legitimate search tasks.

4.4 Optimizing Query Correction Strategies (RSQ3)

To address usability challenges and enhance trust, we investigated user preferences for refining inaccurate search results. Participants suggested several strategies, detailed below, to improve the system's transparency and responsiveness.

4.4.1 Dynamic Filter Adjustment. A participant has proposed an adjustment of the filters, based on the input provided. As illustrated in Figure 6, the idea is to highlight implicitely used filters visually, i.e. dynamically adjusting filters on the interface. Therefore, the user is able to ascertain which filters the AI utilizes and, consequently, identify the potential origin of an error. Therefore, the user has the option of either utilize the filter to resolve the issue or adjusting the initial search query.

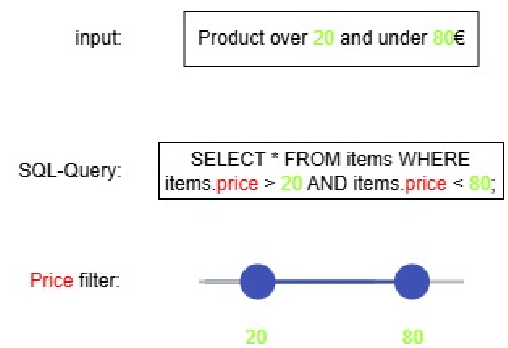

Fig. 6. Here the user can see how the filter is adapted to the user's input. The filter adapts to user input, revealing AI misinterpretations (implicitely).

4.4.2 User Notification for No Results. A further potential avenue for enhancing the efficancy of the user's outcomes is the implementation of an artificial intelligence system that can evaluate the combinability of diverse features during the user's input phase. In such a case, it is essential that the user be alerted to this possibility. The current state of the AI, reflecting the original user input, is shown in Figure 7. The subsequent version has been enhanced to alert the user to the absence of search results for a given combination of features. In this instance, the "under 2 months" feature is distinctly emphasized, as it does not result in any search results. Therefore, the user has the capacity to modify the preliminary search query and discern the elements that are not compatible.

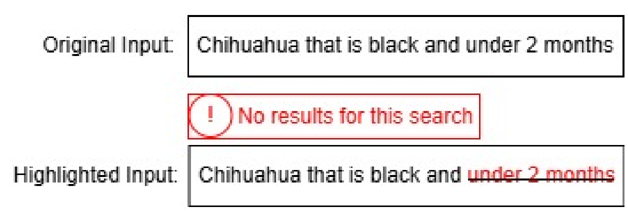

Fig. 7. Here the user is informed by the AI if the features are not combinable. It is made clear that 'under 2 months' cannot be combined with the other features, as this feature is crossed out.

4.4.3 Display of Similar Results. A further point to be considered is the potential for the AI to present analogous products in the event that the search yields few results. A relevant example would be a search for dogs that are of medium size. In the event that the available results are limited, the artificial intelligence could be programmed to display dogs of smaller stature.

4.4.4 Real-time AI-driven Suggestions. Another idea for improving the system is to offer the user search suggestions as they type. This not only enables the AI to deliver more relevant results, but also to better understand what the user is looking for. An example of this is shown in Figure 8. The user enters 'Puppy that is black and...' and the AI suggests narrowing the search - for example to 'Puppy that is black and under three months'.

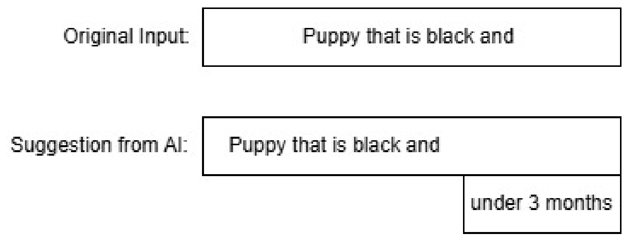

Fig. 8. The AI provides real-time suggestions for narrowing searches, such as additional features for 'Chihuahuas'.

4.4.5 Query Confidence Indicator. Another suggestion from users is that the AI should more clearly indicate its uncertainty when interpreting a query, for example, by providing an uncertainty score ranging from 0 to 1. This score would reflect how confident the AI is in its understanding of the request. This would allow users to better assess whether the provided results might be inaccurate or incorrect, enabling them to make adjustments if needed.

An example from the current AI system illustrates this. When a user searches for a "cheap dog," the AI often simply returns the least expensive dog available. However, this may not align with what the user intended; perhaps they were looking for a cost-effective yet suitable dog for specific needs. An uncertainty score could indicate how reliable the results are, signaling to the user that the interpretation of their query might not be entirely accurate. This would give the user the opportunity to refine their request for more appropriate responses.

5 Discussion

The objective of the paper is to propose the development of an artificial intelligence (AI) system designed to facilitate the search process for products by users.

5.1 Use of Filter

The user study revealed that the majority of users preferred to work with the filters first. This phenomenon may be attributed to the fact that contemporary systems are generally designed to assume that users will employ search filters during the search process. Consequently, users may find it unconventional to utilize solely the search bar to perform a search. The utilization of artificial intelligence by the majority of users does not align with conventional practices, as individuals have become accustomed to employing filters within online shopping platforms rather than utilizing the capabilities of AI directly. Following a series of inquiries, a conclusion was reached: The users have adapted to the system and are now able to discern whether the AI comprehends their intentions. Consequently, the majority of users have increasingly relied on the AI rather than the filters. Therefore, a period of adaptation is required for users to acclimatize to the KI. It is imperative to acknowledge the pivotal role of artificial intelligence in the integration of such systems within e-commerce platforms. The objective is to cultivate a conscious awareness among users.

5.2 Improvement through AI

The implementation of artificial intelligence has yielded notable enhancements in the realm of product search, thereby facilitating a more efficient and user-friendly experience for consumers. The artificial intelligence system demonstrated the capacity to automatically resolve spelling errors, thereby exhibiting enhanced tolerance for errors when compared to conventional systems. A further favorable aspect is that the AI was able to accurately interpret synonyms, thereby identifying the correct products.

5.3 Limitations

The implementation of AI could be exploited by users with relative ease, provided they have some degree of experience with AI. In the user study, for instance, a user effectively replaced the system with the search bar and issued commands through it.

The search function implemented in the conventional system did not align with the prevailing technological standards. The search was exclusively based on textual descriptions, which is a limitation when attempting to emulate existing web shops. It is therefore recommended that a search function be implemented that aligns with contemporary technological standards to facilitate the emulation of modern web shops.

5.4 Future Work

In light of the limited diversity observed in user studies, it would be advisable to undertake a study that involves fewer participants, particularly those with a background in information technology. Additionally, it would be beneficial to include older adult groups to assess the intuitiveness of the artificial intelligence interface. This approach would facilitate a comprehensive evaluation of the ease of use of the AI for different demographics.

A further investigation could be conducted in which the conventional system and the AI are compared on two separate pages. This would allow for the assessment of the efficiency and efficacy of AI utilization in a web shop. This approach could also address the filter bias identified in our user study.

Due to temporal constraints, a static template was employed, with its sole reliance on the description. In this instance, the integration of a state-of-the-art template could potentially enhance the comparison.

Another aspect that must also be mentioned in this context is protecting the system from being exploited by users. It is essential to ensure that users cannot exercise any control over the AI.

5.5 Design Recommendations

6 Conclusions

References

[1] Henry Kautz Ana-Maria Popescu, Oren Etzioni. 2003. *Towards a Theory of Natural Language Interfaces to Databases.* https://dl.acm.org/doi/10.1145/604045.604070

[2] OWASP Foundation. 2024. *OWASP Top Ten | OWASP Foundation.* https://owasp.org/www-project-top-ten/

[3] Eve Rouse. 2023. *New research: 69% of online shoppers go straight to the search bar when visiting ecommerce sites, but 80% leave due to a poor experience.* https://www.nosto.com/blog/new-search-research/