Article

Design of Generalized Search Interfaces for Health Informatics

Jonathan Demelo ![ORCID] and Kamran Sedig *

Department of Computer Science, Western University, London, ON N6A 3K7, Canada; jdemelo8@uwo.ca
* Correspondence: sedig@uwo.ca; Tel.: +1-519-661-2111 (ext. 86612)

Abstract: In this paper, we investigate ontology-supported interfaces for health informatics search tasks involving large document sets. We begin by providing background on health informatics, machine learning, and ontologies. We review leading research on health informatics search tasks to help formulate high-level design criteria. We use these criteria to examine traditional design strategies for search interfaces. To demonstrate the utility of the criteria, we apply them to the design of ONTology-supported Search Interface (ONTSI), a demonstrative, prototype system. ONTSI allows users to plug-and-play document sets and expert-defined domain ontologies through a generalized search interface. ONTSI's goal is to help align users' common vocabulary with the domain-specific vocabulary of the plug-and-play document set. We describe the functioning and utility of ONTSI in health informatics search tasks through a workflow and a scenario. We conclude with a summary of ongoing evaluations, limitations, and future research.

Keywords: information search; search tasks; health informatics; interface design; ontologies; machine learning; PubMed

1. Introduction

Health informatics is concerned with emergent technological systems that improve the quality and availability of care, promote the sharing of knowledge, and support the performance of proactive health and wellness tasks by motivated individuals [1]. Subareas of health informatics may include medical informatics, nursing informatics, consumer informatics, cancer informatics, and pharmacy informatics, to name a few. Simply put, health informatics is concerned with finding new ways to help stakeholders work with health information to be able to perform health-related tasks more effectively. Users in the health domain are increasingly taking advantage of computer resources in their tasks. For instance, a 2017 Canadian survey found that 32% of respondents within their last month had used at least one mobile application for health-related tasks. Even more, those under the age of 35 are twice as likely to do so [2]. Furthermore, studies have calculated that over 58% of Americans have used tools like Google and other domain-specific tools to support their health informatics search tasks—with search being one of the most important and central tasks in most health informatics activities [3,4].

Yet, search can be challenging, particularly for health informatics tasks that utilize large and complex document sets. For such tasks, health informatics tools may require the use of domain-specific vocabulary. Aligning with this vocabulary can be a significant challenge within health tasks, as they can involve a lexicon of intricate nomenclature, deeply layered relations, and lengthy descriptions that are misaligned with common vocabulary. For instance, one highly cited medical research paper defines the term "chromosomal instability" as "an elevated rate of chromosome mis-segregation and breakage, results in diverse chromosomal aberrations in tumor cell populations". In this example, those unfamiliar with the defined term could find parsing its definition just as significant a challenge as the term itself [5]. Thus, when communicating across vocabularies, users may struggle to describe the requirements of their search task in a way that is understandable by health informatics tools [6,7]. To deal with this challenge, ontologies can be a valuable

mediating resource in the design of user-facing interfaces of health informatics tools [8]. That is, ontologies can bridge the vocabularies of users with the vocabulary of their task and its tools. Yet, the use of ontologies in user-facing interface design is not well established. Furthermore, health informatics tools that present a generalized interface, one that can support search tasks across any number of domain vocabularies and document sets, can allow users to transfer their experience between tasks, presenting users with information-centric perspectives during their performances rather than technology-centered perspectives [9,10]. For this, there is a need to distill criteria that can guide designers during the creation of ontology-supported interfaces for health informatics search tasks involving large document sets.

The goal of this paper is to investigate the following research questions:

- What are the criteria for the structure and design of generalized ontology-supported interfaces for health informatics search tasks involving large document sets?
- If such criteria can be distilled, can they then be used to help create such interfaces?

In this paper, we examine health informatics, machine learning, and ontologies. We then review leading research on health informatics search tasks. From this analysis, we formulate criteria for the design of ontology-supported interfaces for health informatics search tasks involving large document sets. We then use these criteria to contrast the traditional design strategies for search interfaces. To demonstrate the utility of the criteria in design, we will use them to structure the design of a tool, ONTSI (ONTology-supported Search Interface). ONTSI allows users to plug-and-play their document sets and expert-defined ontology files to perform health informatics search tasks. We describe ONTSI through a functional workflow and an illustrative usage scenario. We conclude with a summary of ongoing evaluation efforts, future research, and our limitations [11].

2. Background

In this section, we describe the concepts and terminology used when discussing ontology-supported interfaces for health informatics search tasks involving large document sets. We begin with background on health informatics. Next, we examine machine learning. We conclude with coverage of ontologies and their utility as a mediating resource for both human- and computer-facing use.

2.1. Health Informatics

Health informatics is broadly concerned with emergent technological systems for improving the quality and availability of care, promoting the sharing of knowledge, and supporting the performance of proactive health and wellness practices by motivated individuals [1]. Initially, the need for expanded health and wellness services stemmed from rising population levels combined with the growing complexity of medical sciences. These issues made it challenging to maintain quality care within increasingly stressed medical systems [12]. Thus, a central objective for health informatics is the development of strategies to tackle large-scale problems that harm trained medical professionals' ability to perform their tasks in a timely and effective manner. For instance, tele-health services allowed doctors to practice remote medicine, providing care to those without local medical services. Another early innovation was standardized health care records, where patient records were given standardized encodings to provide an increased ability to track, compare, manage, and share personal health information [3]. Some examples of current research directions are the push for stronger patient privacy, personalized medicine, and the expansion of healthcare into under-served regions and communities [1–3,13,14].

The rising production and availability of health-related data has resulted in a growing number of data-intensive tasks within health. Both private and public entities like health industry companies, government bodies, and everyday citizens are turning to health informatics tools as they manage and activate their health data [2]. A growing number of health-related tasks involve searching document sets. During these tasks, the aim of the user is to use the information described within their document set to increase their

understanding of a topic or concept. For example, a search task could be a practitioner searching the electronic health records of their patients, a member of the general public using public materials for their general health concerns, or a researcher performing a literature review [12,15,16]. In general, a search task involves the generation of a query based on an information-seeking objective. The computation systems of these tools then use this query within their computation systems to map and extract relevant documents out from the document set [15]. Powerful technologies like machine learning are increasingly being integrated within tools to help perform rapid and automated computation on document sets [4]. Yet, when taking advantage of these technologies, designers must be mindful of human factors when generating the user-facing interfaces of their tools, as a task cannot be performed effectively without direction from an empowered user [12].

2.2. Machine Learning

Machine learning techniques are increasingly being utilized to tackle analytic problems once considered too complex to solve in an effective and timely manner [16]. Yet, recent analysis [17–19] on the human factors in machine learning environments have found that the current design strategies continually limit users' ability to take part in the analytic process. More so, it has produced a generation of machine learning-integrated tools that are failing to provide users a complete understanding on how computational systems of their tools arrive at their results. This has significantly reduced users' control and lowered the ability to achieve task objectives. In response, there is a growing desire to promote the "human-in-the-loop," bringing the benefits of human reasoning back to the forefront of the design process [20–22].

When considering the interaction loop of a machine learning-integrated tool, Sacha et al. [23] present a five-stage conceptual framework: producing and accessing data, preparing data for tool use, selecting a machine learning model, visualizing computation in the tool interface, and users applying analytic reasoning to validate and direct further use. Assessing this framework, a machine learning-integrated search tool must provide users with a functional workflow where:

1. Users communicate their task requirements as a query.
2. Users ask their tool to apply that query as input within its computational system.
3. The tool performs its computation, mapping the features against the document set.
4. The tool represents the results of the computation in its interface.
5. Users assess whether they are or are not satisfied with the results.
6. Users restart the interaction loop with adjustments or conclude their use of the tool.

Thus, a primary responsibility for users within machine learning environments is the need to assess how well the results of machine learning have aligned with their task objectives. A systematic review by Amershi et al. [24] suggests six considerations for the user's role in arbitrating machine learning performance:

1. Users are people, not oracles (should not be expected to repeatedly answer whether a model is right or wrong).
2. People tend to give more positive than negative feedback.
3. People need a demonstration of how machine learning should behave.
4. People naturally want to provide more than just data labels.
5. People value transparency.
6. Transparency can help people provide better labels.

2.3. Ontologies

In search tasks involving large document sets, many challenges can arise that reduce performance quality, harm user satisfaction, and increase the time for task completion [17–19]. Often, these challenges result from misalignment between the vocabularies used by the document sets, storage maintainers, interface designers, and users. For instance, Qing et al. [25] outline the difficulties faced when translating between common and domain vocabularies in health tasks. They describe a study that found that up to 50% of health expressions by

consumers were not represented by public health vocabularies [25]. Within the pipeline of a search task, both the human and computational system can only perform optimally if communication is strong [26]. Ontologies are representational artifacts that reflect the entities, relations, and structures of its domain. Ontologies are of three types: a philosophical ontology for describing and structuring reality, a domain ontology for structuring the entities and relations of a knowledge base, and a top-level ontology for interfacing between different domain ontologies [26]. Ontologies provide the flexibility, extensibility, generality, and expressiveness necessary to bridge the gap when mapping domain knowledge for effective computer-facing and human-facing use [8]. For this purpose, ontologies are increasingly being used within tools to help users perform their challenging search tasks [27–31].

When creating an ontology, experts construct a network of entities and relations, which together yield various structures [32,33]. Ontology entities reflect the objects of the domain, like a phenotype in a medical abnormality ontology, a processor in a computer architecture ontology, or a precedent in a legal ontology [34]. In some ontologies, like the top-level ontology, Basic Formal Ontology, designers go as far as denoting qualities such as materiality, object composition, and spatial qualities in reality [26]. Ontology entities are encoded with information about their role in the vocabulary, definitions, descriptions, and contexts, as well as metadata that can inform the performance of future ontology engineering tasks. Ontology relations are the links between ontology entities that express the quality of interaction between them and the domain as a whole [35]. When assessing ontology relations, Arp et al. [26] distinguish relations under the categories of universal–universal (dog "is_a" animal), particular–universal (this dog "instance_of" dog), and particular–particular (this dog "continuant_parts" of this dog grouping). Domain ontology relations are realized through unique interoperability between ontology entities. For instance, an animal ontology may have an ontology entity reflecting the concept of a "human," which may have the ontology relations "domesticates/is domesticated by" between it and the "dog" ontology entity.

After defining the entities, relations, and other features of an ontology, experts record their work in ontology files of standardized data formats like RDF, OWL, and OBO. These ontology files are then distributed amongst users. They can then be integrated into the computational and human-facing systems of tools for use during tasks. Some examples of current ontology use are information extraction on unstructured text, behavior modeling of intellectual agents, and an increasing number of human-facing visualization tasks such as decision support systems within critical care environments [20–22].

3. Methods

In this section, we describe the methods used for criteria formulation. We begin with a review of literature for health informatics search tasks. Based on the insights gained from this review, we distill a set of criteria. We then use these criteria to contrast traditional design strategies for interfaces of search tasks.

3.1. Task Review

Here, we review some research on interfaces for health informatics search tasks. We used Google Scholar, IEEE Xplore, and PubMed to conduct an exhaustive search of articles and reviews published between 2015 and 2021. We have divided our findings into three sections. First, we explore research on health data, information management, and information-centric interfaces. This is followed by research discussing the types of search tasks and their use in structuring the design of interfaces for health informatics. Finally, we investigate the requirements for aligning vocabularies for health informatics search tasks.

3.1.1. Health Data, Information Management, and Information-Centric Interfaces

Health data is constantly generated, highlighted by reports that within just a year the US healthcare system created 150 new exabytes of data [36]. Yet, the information that

is expressed by this data, such as personal medical records, research publications, and consumer health media, is not useful unless it can be effectively understood and utilized by users. As such, it is critical to examine the challenges facing users when performing their tasks, and through this establish novel strategies for supporting the activation of health data.

Fang et al. [10] explore the pressing challenges for accessing health data under the four categories: volume, variety, velocity, and veracity. They find that the volume of health care data creates challenges in the management of data sources and stores. They describe that existing strategies are struggling, and that novel designs should be established for scaling data services. They explain that a variety of challenges come with the management of data characteristics, ranging from unstructured datapoints generated from sources like sensors, to structured data entities like research papers and medical documents. For this, they state that designers should concentrate on aligning with the characteristics of the information being encountered. Next, they explore the challenges of velocity, which involves the rate at which users require their data to move from source to activation within their task. They highlight novel research in the networking and data management space. Finally, they explore veracity challenges, such as the assessment and validation of data quality and the quality of information that the data may produce.

Gibson et al. [9] provide a review of the evolving fields of health information management and informatics. They review the topics of data capture, digital e-record systems, aggregate health management, healthcare funding models, data-oriented evidence-based medicine, consumer health applications, health governance, personal health access, and genomic personalized medicine. Similar to Fang et al. [10], they note that the predominant work for health informatics should be concerned with presenting users with information-centric perspectives during their performances rather than technology-centered perspectives. More so, they describe that users in healthcare "must often navigate and understand complex clinical workflows to effectively . . . capture, store, or exchange information". In other words, task workflows are already complex; therefore, effective interfaces should promote information encounters that help users perform better, rather than engage in unrelated technical details.

From the above research, we distill the criterion: Designs should maintain an information-centric interface that is flexible with respect to the dynamic requirements of search tasks like veracity of data sources, variety of data types, and evolving needs of users for health informatics.

3.1.2. Search Tasks and Structuring the Design of Interfaces for Health Informatics

Russell-Rose et al. [37] describe professional health workplace tasks. They find that the most prevalent types of search tasks are literature reviews for overviewing a topic, scoping reviews for rapidly inspecting the possible relevance of an information source, rapid evidence reviews for appraising the overall quality of a scoping review, and, finally, systematic reviews for exploring a topic in a robust manner.

During search tasks, users often lack the ability to perceive how their query decisions impact, relate, and interact with the document set. This is an important consideration for users who might want to adjust a query to better align with their information-seeking objectives. A further study by Russell-Rose et al. [38] analyzes search strategies performed by healthcare professionals. They find that a large majority of participants have a general desire to utilize advanced search functionalities when available. This suggests that users are not hesitant to take advantage of resources that they believe help optimize their task performance. Huurdeman [39] outlines that for this, a good course of action is to leverage query corrections, autocomplete, and suggestions. Yet, they find that such additions can be harmful if those features do not provide appropriate domain context. That is, resources must allow users to be contextually aware of how their query aligns with the contents of document sets, as well as the conditions of computational technologies used by interfaces.

In the same research, Huurdeman [39] investigates complex tasks involving information search and information-seeking models when using multistage search systems. In this

research, they explore requirements that designers must account for when supporting users. Challenging search tasks require users to learn about the searched domain, understand how their objectives align, and formulate their objectives into a way that can be used by their tool. In other words, query building requires users to be domain cognizant, as they must communicate information-seeking objectives in a way that is understood by the tool, yet also aligns with the information found within the document sets. Thus, a health informatics tool that supports search tasks should provide the opportunity for understanding the domain of the document set being explored.

Zahabi et al. [40] describe a set of nine requirements for designers when considering how to design usable interfaces for health informatics search tasks, summarized as:

- Naturalness: The workflow of the system must present a natural task progression.
- Consistency: The parts of the system should present similar functional language.
- Prevent errors: Be proactive in the prevention of potential errors.
- Minimizing cognitive load: Align cognitive load to the requirements of the task.
- Efficient interaction: Be efficient in the number of steps to complete a task.
- Forgiveness and feedback: Supply proper and prompt feedback opportunities.
- Effective use of language: Promote clear and understandable communication.
- Effective information presentation: Align with information characteristics.
- Customizability/flexibility: The system should remain flexible to the task requirements.

Additional research by Dudley et al. [41] reviews user interface design for machine learning environments. They provide a set of principles that can be used by designers:

- Make task goals and constraints explicit.
- Support user understanding of model uncertainty and confidence.
- Capture intent rather than input.
- Provide effective data representations.
- Exploit interactivity and promote rich interactions.
- Engage the user.

From this research, we can distill two criteria: First, designs should provide interaction loops that promote prompt and effective feedback opportunities for the user. Second, designs should provide representations that are natural and consistent to the requirements of the information source, the user, and the task.

3.1.3. Aligning Vocabularies for Health Informatics Search Tasks

When considering interfaces for health informatics search tasks, a major challenge for users is the need to overcome problem formulation deficiencies when encountering unfamiliar domains. This is because, according to Harvey et al. [42], users have been found to consistently suffer from four major issues during the performance of search tasks:

- Difficulty understanding the domain being searched.
- An inability to apply their domain expertise.
- Lacking the capacity to formulate an effective search query within the interface that accurately reflects their information-seeking objective.
- Deficient understanding of how to assess results produced by search, to decide whether the search has or has not satisfied their objective.

Harvey [42] shows that in domains with complex vocabularies, such as health and medicine, the disparity of potential users' prior knowledge is extreme. They find that non-expert users routinely do not possess enough domain knowledge to address their information-seeking needs. This can cause significant issues during query formulation. As a result, non-expert users must first step away from their tool to learn specialized vocabulary before they can begin query building. Both Soldaini and Anderson [43,44] describe that this issue can still affect even experts. This is because experts often must make assumptions when attuning to their tool.

There is growing research targeting the generation and application of mediation resources to help reduce the communication gap while using health informatics tools.

Zeng et al. [25] investigate the development of consumer health vocabularies for reducing the discourse gap between lay people and medical information document sets. Furthermore, Soldaini et al. [45] explore the use of novel query computation strategies to improve the quality of medical literature retrieval during search tasks. In their quantitative study, they contrast models generated using combinations of algorithms, vocabularies, and feature weights, assessing the computational performance of different query reformulation techniques. The results of their study suggest "greatly improved retrieval performance" when utilizing combined machine learning and bridged vocabularies. More so, they provide insight regarding the quality of options that can support computational systems for health informatics search tasks.

From the above research, we can distill two criteria. First, designs should provide interactions that allow users to efficiently prepare, perform, assess, and adjust their machine learning to align with information-seeking objectives of search tasks. Second, designs should provide mediation opportunities that assist users in communicating information-seeking objectives into the domain-specific vocabulary of the document set.

3.2. High-Level Criteria

In Table 1, we provide five criteria based on the above review.

Table 1. The criteria for guiding the design of interfaces for health informatics search tasks involving large document sets. For abbreviation purposes, design criteria will be referenced in the text as DC#, where # is its assigned number.

DC#	Design Criteria
DC1	Provide an information-centric interface that shows flexibility towards the evolving needs of users and the dynamic requirements of search tasks like the veracity of data sources and variety of information types.
DC2	Provide interaction loops that supply prompt and effective feedback for users during the performance of search tasks.
DC3	Provide natural and consistent representations that allow users to understand the constraints, processes, and results provided by the interface.
DC4	Provide interactions that allow users to efficiently prepare, perform, assess, and adjust their machine learning to align with the information-seeking objectives of search tasks.
DC5	Provide mediation opportunities that assist users in communicating and bridge their information-seeking objectives into the vocabulary of the document set.

3.3. Analysis of Traditional Interface Strategies for Health Informatics Search Tasks

We now assess the traditional design strategies for interfaces for health informatics search tasks. Wilson's comprehensive Search User Interface Design [46] provides a complete survey of the history and current state of search interfaces. Based on their survey, and in particular their discussion of input and control features within the modern search user interfaces, two base strategies and one extension strategy for search interfaces are realized: "structured" interfaces, "unstructured" interfaces, and, in extension, "query expansion" interfaces. Table 2 provides a summary of how the above criteria align with each interface strategy.

3.3.1. Structured Interface Strategy

The structured interface strategy creates designs that regulate input during query building. This is achieved by maintaining heavily restricted input control profiles. Designers who implement the structured interface strategy into their interfaces presuppose a search task with specific expectations for input, bounding queries to a limited input profile. One common bounding technique is to constrain query lengths and limiting query content to a controlled set of terms [47]. This restricted scope is considered the sole acceptable input profile, and thus allows designers to generate interfaces that limit the possible range of inputs and restrict all inputs that fall outside of that range. Designers typically achieve

this by using interface elements like dropdowns, checkboxes, and radio buttons instead of elements like text boxes with free typing. For example, Figure 1 depicts the PubMed Advance Search Builder, which implements the structured interface strategy in its design. This interface requires users to select specific query term types from a restricted list, which then guides user input [48].

Table 2. A summary matrix of alignment between the criteria and interface strategies. Full descriptions are found within their respective sections. "Strong" is assigned if a characteristic of the interface strategy promotes alignment with the requirements of the design criterion. "Weak" is assigned if a characteristic of the strategy does not promote alignment with the requirements of the design criterion. "Variable" is assigned if the interface strategy has the potential to align with the criterion; however, such an alignment is not innate and must be actively pursued.

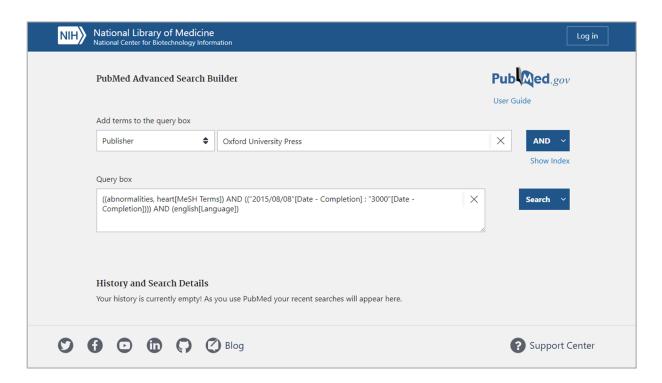| DC#	Structured	Unstructured	Query Expansion
DC1	Weak	Variable	Strong
DC2	Strong	Strong	Variable
DC3	Variable	Variable	Variable
DC4	Variable	Strong	Strong
DC5	Weak	Weak	Strong

Figure 1. PubMed Advanced Search Builder: an example of a structured interface strategy for a search task. In this use case, a query item was generated for a MeSH term for heart abnormalities, a completion date after 8 August 2015, and in the English language, with the publisher of Oxford University Press soon to be added. Source: Image generated on 18 January 2021, using the public web portal provided by the National Center for Biotechnical Information, https://pubmed.ncbi.nlm.nih.gov/advanced/ (accessed on 18 January 2021).

Since input control is restricted, a strength of the structured interface strategy is that designers can use information characteristics to prescribe the full range of query formulations. This allows for the use of representational and computational designs that optimize for the expected characteristics of the restricted input profiles, per DC2. This strategy provides a designer-friendly environment that is hardened against unwanted queries, which, if effectively communicated in the design of result representations, could allow for alignment with DC3 and DC4. Yet, it can be challenging to designers to use

structured interface strategies in a generalized setting. This is because when a document set is swapped, hardened approaches may not align with the information characteristics of the new document set. This negatively affects the flexibility of the interface, and in turn alignment with DC1. A potential weakness of the structured interface strategy is that it requires users to possess expertise on both the controlled vocabulary of the interface as well as the vocabulary of the document set being searched. If this is not known, user experience can suffer, drastically affecting alignment with DC5. Within the context of health informatics, such weaknesses reduce the users' ability to effectively perform search tasks. This is because the controlled vocabularies within the health and medical domains demand significant expertise and result in numerous points of failure during the query formation process [49,50].

3.3.2. Unstructured Interface Strategy

The unstructured interface strategy creates designs that provide limited input regulation. Unlike the structured interface strategy, it provides an open input control that accepts most input profiles during query building. Designers who implement the unstructured interface strategy do so without presupposing particular input, only accounting for general user error. That is, this input can originate from anywhere, such as common vocabulary, rather than from a pre-determined set of terms provided by the designer. Often, this input is directed to a single interface element. Implementations of the unstructured interface strategy typically present a text box that allows users to freely type their own text into the interface. These implementations will perform some input processing prior to use; however, the presentation of this processing to users is usually limited to correcting typographical errors rather than semantic ones. For example, the interface of Google aligns with the unstructured interface strategy, presenting users with an open, text-box input control without domain-specific assumptions or requirements. Of course, Google's computational systems use extensive processing between receiving input from users and presenting the results of computation back to users [51]. Yet, users themselves are not informed of how their results came to be, even after changing to Google Instant [52]. Another example of an implementation of the unstructured interface strategy is WebMD's search interface. This interface processes a free-text input with basic sanitization techniques before generating features for its search engine system, as depicted in Figure 2.

A strength of the unstructured interface strategy is that it supports the use of any vocabulary during query building, allowing for the natural activation of common vocabulary during task performance, in alignment with DC4. Additionally, this removes the requirement for users to possess input expertise and control profiles that typically come with a structured interface strategy, per DC1 and DC4. Designers can still implement prompt and effective feedback during task performance, thereby supporting DC2. If the constraints, processes, and results of their task performance are effectively communicated in result representations, DC3 and DC4 can be well supported. However, by allowing for the direct use of common vocabulary in lieu of a presupposed controlled vocabulary, the unstructured interface strategy suffers where the structured interface strategy excels. That is, poor implementations of the unstructured interface strategy can produce interfaces that do not provide mediation for users to translate their common vocabulary into the domain-specific vocabulary. In doing so, users are not being helped in understanding how their query building has impacted their search performance. For example, these poorly implemented interfaces may take input literally and bring users directly to a result page without providing context as to how the results were found, negatively affecting DC1 and DC5. This potential for promoting weak alignment between user and information source can lead to a significant drop in the quality of search performance. This can be an especially important requirement to address for health informatics interfaces, as it has been found that users routinely struggle to craft effective query terms during their health-related search tasks [53].
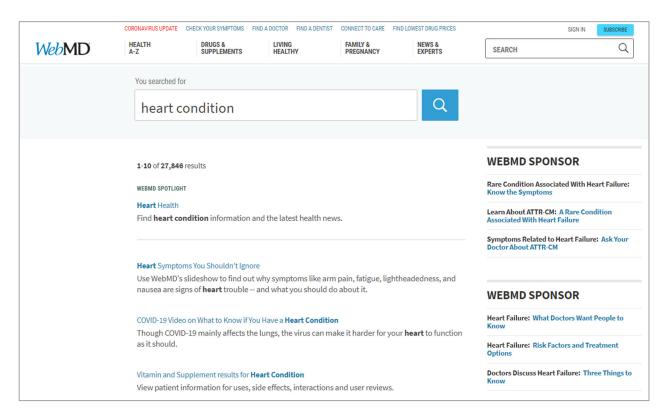
Figure 2. WebMD Search Interface: an example of an unstructured interface strategy for a search task. In this use case, the free–text query "heart condition" was generated. Source: Image generated on 18 January 2021, using the public web portal provided by WebMD, https://www.webmd.com/search/search_results/default.aspx?query=heart%20condition (accessed on 18 January 2021).

3.3.3. Query Expansion Interface Strategy

The query expansion interface strategy is an extension of both the structured and unstructured interface strategies. That is, this strategy expands by adding mediation opportunities to bridge the vocabulary of the user with the vocabulary of the document set both within the representational as well as the computational systems [54]. These mediation opportunities are typically implemented within two parts of the interaction loop. The first is during input, where mediating opportunities present during query building. Often, these mediation opportunities come as cues that suggest to users how their common vocabulary could align with the vocabulary of the domain, and visa-versa. An example of an implementation of the structured-like query expansion interface strategy is WebMD's Symptom Checker, shown in Figure 3. This example interface goes through a series of controlled stages of query building that are structured by numerous opportunities for mediation. The second is during the processing prior to document set mapping. Like other strategies, a system can apply natural language processing techniques to the input, where the text string provided as input is tokenized into its parts. From this, the system sanitizes token parts to remove trivial tokens like the stop words "the," "a," and "an," and any remaining tokens are then inserted as features in search engine systems. In more complex systems, additional sanitization techniques can be used [55]. Yet instead of immediately inserting the remaining tokens as features into the computational systems, the query expansion interface strategy builds upon the input profile by injecting insight provided by mediating resources, such as related terms, synonyms, and other expansion opportunities [56]. In other words, these systems utilize mediating resources to computationally expand the query. Some examples of mediating resources are knowledge bases like WordNet and Wikipedia, and ontologies like The Human Phenotype Ontology [11,54,57].
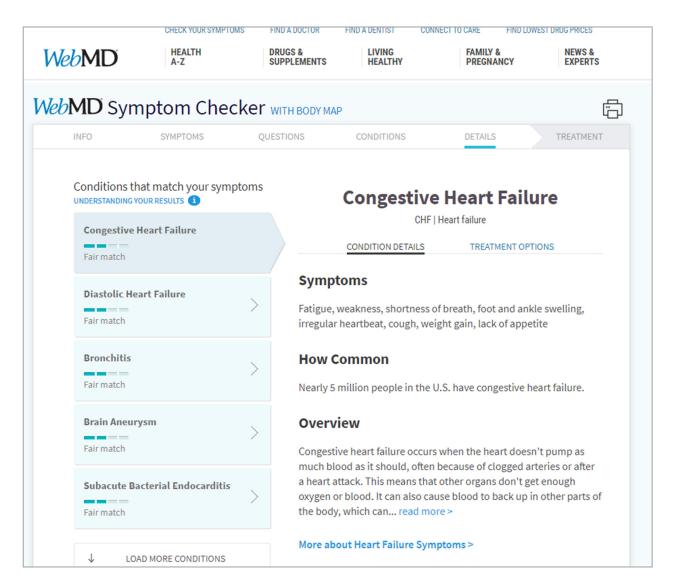
Figure 3. WebMD's Symptom Checker: an example of a structured–like query expansion interface strategy for a search task. In this use case, users are guided along a series of query building opportunities, allowing them to enter various symptoms and personal health criteria while aligning their personal vocabulary with the information resource vocabulary. Source: Image generated on 18 January 2021, using the public web portal provided by WebMD, https://symptoms.webmd.com/ (accessed on 18 January 2021).

A strength of the combined approach of the query expansion interface strategy is its strong efforts to eliminate the weaknesses associated with the structured and unstructured interface strategies while still maintaining their strengths. That is, by allowing the continued use of common vocabulary during the process of query building, users can have higher confidence about what the interface is asking of them, and what they are telling the interface to do, helping with DC4 and DC5. Furthermore, by integrating the use of mediating resources like ontologies, designers can demonstrate to users the quality of their query building and how their vocabulary decisions affect the performance of their search tasks, supporting DC2 and DC3 [58]. Yet, with the added complexities of query expansion, computational systems may be required to perform more work before arriving at a final set of search results. Therefore, designers of systems taking advantage of query expansion should consider the impact on performance and responsiveness and counteract them to maintain alignment with DC2. For the query expansion interface strategy to be successful, designers must clearly communicate to users how exactly their query building has affected their search. If this communication is not provided, it can leave

users confused regarding how their decisions have affected their search and can make it challenging for them to assess task performance, negatively affecting DC2. Such limitations may not provide optimal alignment in communication between the system, the user, and the information resource [53]. That is, if a selected mediating resource does not provide an effective mapping between vocabularies, then query expansion can weaken the quality of search tasks. To address this challenge, designers can utilize user-supplied ontologies, as per DC1. This provides users the freedom to select mediating resources that they believe can best support their task performance, rather than being restricted to a tool-provided mediating resource. A user study by Jay et al. [59] compares users as they perform the same task set using two interfaces, one with a structured multiple variable input profile, the other with an unstructured single variable input profile. In this study, they find that users felt their needs and expectations were better fulfilled using the single-input profile, performing their tasks quicker, with more ease of use and learnability, and with a higher appraisal of results. Designers must carefully select how they activate query expansion such that it addresses the needs of the task, the information, and the user.

4. Results

In this section we describe ONTSI, a generalized ontology-supported interface for health informatics search tasks involving large document sets created using the above-discussed criteria. We outline how the criteria were used to structure ONTSI's design. We then discuss the technical scope of ONTSI, concluding with ONTSI's functional workflow.

4.1. Design Scope

Table 3 highlights the role of each criterion in the design of ONTSI.

Table 3. The role of each criterion within the design of ONTSI. The incorporation of these criteria in ONTSI's implementation is discussed within the workflow and usage scenario.

DC#	ONTSI
DC1	ONTSI leverages powerful third-party computational technology. Specifically, pre-built machine learning packages like SciKit-Learn are integrated within ONTSI, and highly optimized indexing is provided by The Apache Software Foundation's Solr product [60]. Additionally, ONTSI's interface provides users with clear text-based alerts, which reflect their current performance status.
DC2	ONTSI supports an iterative interaction loop to allow users perform repeated sets of search tasks. That is, within iterative interactions, users can save the results they regard relevant in a persistent location within the tool, while still allowing further performances to occur.
DC3	ONTSI provides visual representations to help analyze and judge the relevance of search results.
DC4	ONTSI utilizes modern visualization and computational technologies like D3.js to provide powerful interaction opportunities.
DC5	ONTSI supports the use of a common vocabulary during query building using the query expansion strategy. Specifically, when using ONTSI, users upload both a document set and an ontology file, which are then integrated into the workflow of the computational systems of ONTSI. Users can interact with a search textbox that allows for unstructured text input. ONTSI provides domain-specific vocabulary suggestions that can assist users in guiding their performance and promote alignment between their vocabulary and domain-specific vocabulary.

4.2. Technical Scope

ONTSI is developed as a web-based tool that provides a generalized, plug-and-play support of user-supplied ontology files and document sets. That is, ONTSI allows for the uploading of ontology files, either individually or within a .zip compressed file, as well as any compressed document set in the ZIP format. ONTSI then processes and indexes their contents for use within the interface. ONTSI's front end uses the latest HTML5, CSS,

and JavaScript technologies, allowing for cross-browser (i.e., Firefox, Chrome, Opera) and cross-platform support. The D3.js JavaScript library is used to create the visualization and interaction experiences found throughout the front end of ONTSI [61]. ONTSI's back-end technology is developed using a custom Python-based computational server that maintains data transfer and machine learning APIs, and with the use of Apache's Solr system as the search indexer and engine [60]. The current ONTSI system maintains support for the live uploading of well-formed ontologies in the Ontology Web Language (OWL) format.

4.3. Functional Workflow of ONTSI

ONTSI encompasses several subsystems and subviews within its workflow. Recalling the workflow description of a machine learning-integrated search tool, ONTSI allows:

1. Users to communicate their task requirements as a query within its Upload and Search subview.
2. Users to ask their tool to apply that query as input within its computational system within its Search Subview.
3. The tool to perform its computation, mapping the features against the document set within its ONTSI server and Solr server.
4. The tool to represent the results of the computation in its interface within its Result List and Result Item subviews.
5. Users to assess whether they are or are not satisfied with the results within its Result List, Result Item, and Saved List subview.
6. Users to restart the interaction loop with adjustments within the Upload and Search subviews or conclude their use of the tool.

We will now describe the overall functional workflow of ONTSI and its parts, as depicted in Figure 4.

4.4. Front-End Subviews

ONTSI consists of a series of interconnected subviews, shown in Figures 4 and 5. We will now describe the functional workflow of each subview.

4.4.1. Upload Subview

The Upload subview supports the plug-and-play of user-supplied ontology files and document sets. This subview can be found at the top left of ONTSI, Figure 5a. When clicked, the upload button opens a file selection window. The window limits uploading to valid ontology files under the OWL ontology format and the .zip compression format. When a compressed file is uploaded, it is inspected for OWL files. This allows the upload system to not only take in individual OWL ontology files, but also sets of OWL files that are combined in a compressed format. Ontology file contents are put through a custom OWL to JSON processor, and then indexed into a local storage system within the browser memory. If it is a document set, it is transferred to the back end ONTSI server. Once at least one ontology file and one document set are uploaded, the Search subview and the system become active.

4.4.2. Search Subview

The Search subview facilitates query building using an ontology-supported unstructured-like query expansion strategy. Three points of interaction are maintained: Query Input, the Run button, and the Clear button. The Search subview is located to the right of the Upload subview at the top center, Figure 5b, and becomes available for interaction after the requirements of the Upload system are fulfilled.

Query Input is a text input box. As text is typed, ONTSI cross-references that text against the uploaded ontological content for mediation opportunities. If found, those mediations are provided within an expanding dropdown. When a user values a suggested mediation, it can be selected and locked in as a query term. If none are desired, they can be ignored. When a user is satisfied with their own typed text, it can also be added. Each

query term is depicted with the text of the term and a removal interaction, represented by a trailing "x" button. If multiple terms require removal, this can be done either with individual removal actions, repeated backspacing actions from the keyboard, or the red "trash can" button, which clears all query terms.

Figure 4. Depiction of the functional workflow of ONTSI. Labelled arrows reflect the steps of the interaction loop. Brown boxes represent the processes performed within the back-end computation systems. Blue boxes represent the object types that persist within the browser and external index database storage. Pink boxes represent the back-end computational systems of ONTSI. Yellow boxes represent the various subviews within the front end of ONTSI. The green box represents the types of interactions that can be conducted with the system.

When at least one query term has been entered, the green "Run" button becomes active. This initiates the performance of computation on the uploaded document set using the ontology file for query expansion. Query terms are collected and sent to the back end ONTSI server system. The Result List subview updates when the computation is complete.

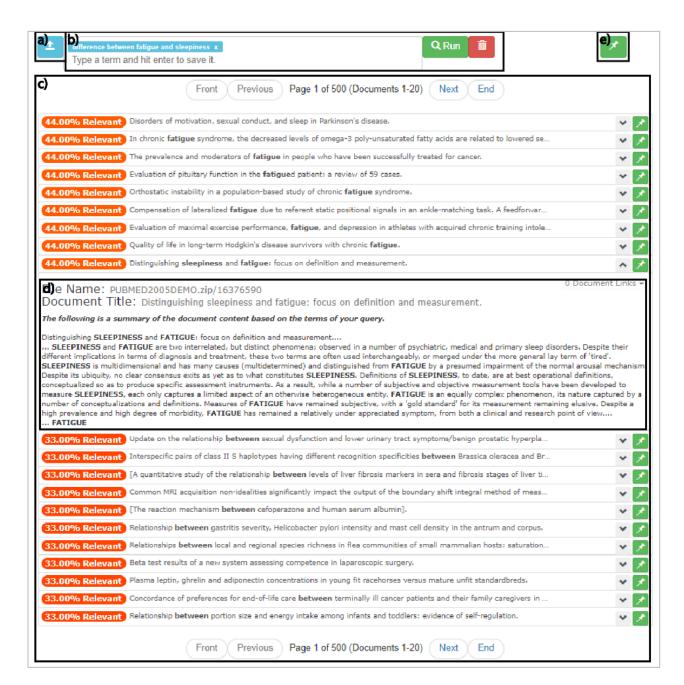**Figure 5.** The overall view of ONTSI in full use and outlined coverage of its five subviews: Upload subview, partial (**a**); Search subview (**b**); Result List subview (**c**); Result Item subview (**d**); and Saved List subview, partial (**e**).

4.4.3. Result List Subview

The Result List subview provides a paged listing the search results. The Result List subview is found directly under the Search, Upload, and Saved List subviews, Figure 5c.

Once a search is performed, the Result List subview changes from an informational alert to the results of a search. The list itself is bounded above and below by buttons and text that describe and support paging interactions. Specifically, the buttons and text describe information about the current page position, the number of pages used to divide the document set, and the number of documents in the current page, and allow for various navigation interactions on the pages.

The search results are sorted by their relevance calculation generated during clustering, such that the results assigned to document clusters that have the highest predicted relevance rating are prioritized. Then, the list is paged. Instantaneous navigation between pages

is provided. Color-coded relevance ratings accompany each document, ranging from best to worst within a green–red color spectrum. Each result represents the document title with annotations highlighting terms or phrases that are believed to align with the provided query terms. A button is also provided that allows the user to access additional document content and open the document for deeper inspection. Finally, each result has a "pin" button, which allows for the saving of documents for future use within the Saved List subview.

4.4.4. Result Item Subview

The Result Item subview provides document-level information. This allows users to rapidly assess the content of individual documents during their search task. When a user selects a document within the Result List subview, ONTSI will request the full document content of that result from the Solr server using its HTTP-based API. Query terms are then used by annotation services within the Solr server to wrap HTML-based annotation tags into the document content, which is then returned to the Result Item subview. When a document is selected for inspection, the Result Item subview expands that document in place within the Result List subview, pushing down trailing items, Figure 5d.

The content of the selected document is represented in the following order: the file name of the document within the uploaded document set, the full document title, and a summarized version of the document content. The summarized version of the document content restricts the document to the passages of content that surround or have associations with the query terms provided during query building. Terms are highlighted through capitalization and with bolded font. In addition, the Result Item subview provides a dropdown at the top right, which collects all web links found within the document content for quick access. Any number of documents can be opened within the Result Item subview for comparison.

4.4.5. Saved List Subview

Each result within the Result Item subview includes a green "pin" button, which saves documents for future reference. ONTSI collects these saved documents within the Saved List subview. The Saved List subview can be accessed at the top right of ONTSI's overall view, directly to the right of the Search subview. There, a green "pin" Saved List can be found that allows us to request ONTSI to open the Saved List modal, Figure 5e. Upon request, the Saved List modal displays saved documents. Here, documents can be recalled, removed from the list, or copied for external use.

4.5. Back-End Systems

ONTSI consists of two back-end systems that support the various front-end subviews and their controlling logic: the ONTSI server and the Solr server. Through their use, heavy computation is moved away from the browser and into dedicated computational systems. This allows for a reduction in computational overhead within the browser to improve response times and allows ONTSI to access computational technology that is not readily supported in the browser.

4.5.1. ONTSI Server

The ONTSI server is created using the Python-based Flask framework. It exposes an API supporting communication between the various systems of ONTSI. The API satisfies two major roles: preparing the uploaded document set for indexing within the Solr server and handling machine learning requests for search tasks.

When a document set has been signaled for upload within the Upload subview, it is packaged and sent through the API of the ONTSI server. Incoming document sets are assessed and provided a suitable decompression algorithm. Next, for each document within the document set, the ONTSI server assesses the encoding of that file (e.g., UTF-8, UTF-16, PDF, etc.). Based on this assessment, a suitable transcription algorithm is applied to

that document. The indexing process for the Solr server is a pull interaction, so documents are stored in a static location from which they can be pulled. Therefore, the documents are sanitized, packaged, and then inserted into a temporary PostgreSQL database. The ONTSI server then requests the Solr server to begin indexing the new document set.

When a search task is initiated, the request is sent to the API of the ONTSI server. There, requests are read for settings like the clustering algorithm, the document set being searched, and query specifications. The ONTSI server then prepares the machine learning environment. Next, ONTSI performs query expansion. This involves a set of natural language preprocessing steps on the query and its individual query items, such as tokenization and the application of stop word limiters. Then, each query item is examined against the provided ontology file for mediating opportunities alongside a complete synonym ring analysis on each query item using WordNet. The original query terms and their associated ontology and synonym terms are then packaged together. These packages are then applied during the performance of unsupervised K-means clustering computation from SciKit-Learn, a third-party machine learning suite. The computed weighting characteristics of clusters are then propagated back as a package of clusters and their associated documents for the ONTSI front end for use within its various subviews. We include a pseudocode representation of these steps in Figure 6.

Algorithm 1: CLUSTERING pseudocode between ONTSI, ONTSI Server, and Solr Server

Input: A set Q of user inputted queries
Output: Signal to update interface with cluster assignments

1   $targets \leftarrow chain(Q).unique().difference(getStopWords())$
2   $documents \leftarrow getDocuments()$

    /* Prepare bag of words using target, related entities, and
      their generated WordNet synsets   */

3   **for** $i = 0$ **to** $targets.length$ **do**
4      $target = targets[i]$
5      $targetCoverage \leftarrow target + target.getDirectlyRelatedEntities()$
6      $targetSpread[target] \leftarrow$
       $targetCoverage + targetCoverage.getWordNetSynsets()$
7      $targetSpread[target] \leftarrow$
       $targetSpread[target].unique().difference(getStopWords())$

    /* Gather counts from pre-indexed documents, then fit and
      predict clusters using Scikit.Learn KMeans clustering   */

8   $documentCounts \leftarrow$
   $getIndexesFromSolrAPI(targetSpread, documents).scaleRange(0, 1)$

9   $reducedPCA \leftarrow SciKitLearn.PCA(nComponents =$
   $2).fit\_transform(documentCounts)$
10   $kmeansPCA \leftarrow SciKitLearn.KMeans(init =' k\text{-}$
   $means + ', nClusters = 7, nInit = 10)$
11   $clusterAssignments \leftarrow kmeansPCA.fit\_predict(reducedPCA)$
12   **for** $i = 0$ **to** $targets.length$ **do**
13      $target = targets[i]$
14   **for** $j = 0$ **to** $clusterAssignments.length$ **do**
15         $cluster = clusterAssignments[j]$
16         $yPred = cluster.yPred[target]$
         /* Generate weighting scale using x5 multiplier   */
17         $clusterAssignments[j].weighting[target] \leftarrow$
         $generateClusterWeighting(yPred)$

18   **return** $updateInterface(clusterAssignments)$

Figure 6. Pseudocode of clustering spanning the workflow of ONTSI (front end), ONTSI server, and Solr server.

4.5.2. Solr Server

ONTSI uses Solr, a third-party document indexing software developed by The Apache Software Foundation. Solr is a scalable indexing system that provides a valuable array of features like a REST-like API supporting many HTTP-based communication interfaces. Solr also provides and a wide range of customizable settings and schemas that supports any number of storing, searching, filtering, analysis, optimization, and monitoring tasks. For a more information regarding Solr and its various permutations, seek out their official website and documentation [60].

A cloud-based permutation of the Solr server is used to handle the indexing and serving of uploaded document sets. Indexing occurs when a request is made to the Solr server from the ONTSI server. The Solr server schema will seek out the location of the temporary PostgreSQL database hosted by the ONTSI server, extract all new documents not already indexed, and apply a processing schema on those documents for indexing. Then, signals are sent out to the relevant ONTSI systems. Solr also handles serving requests when ONTSI requires document content, either at the metadata level when loading the Result List subview, or full, annotated document content in the Result Item subview. Requests are communicated to the Solr server through its HTTP-based API. The Solr server then handles the request, packages the results under the conditions specified in the request, and returns its response.

5. Usage Scenario

In this section, we provide a health informatics search task scenario using ONTSI. We begin with a description of the user profile, as well as the ontology file and document set in the usage scenario. We then present the usage scenario.

5.1. User Profile

The user profile we select here is that of a health stakeholder, a researcher within a professional workplace setting performing a scoping review as an information-seeking objective. A scoping review is concerned with establishing an initial idea of the amount of information on a topic within a document set [37]. The user has a general level of knowledge, typical of other health stakeholders. For instance, the user understands and can communicate phenotypic abnormalities like a broken leg, light-headedness, or loss of vision. The user understands how to perform typical actions on the interface like clicking, typing, and saving, but does not possess knowledge of the technical concerns typical of backend computational technologies.

The objective of the user is to learn whether there are any documents within a document set that are relevant to a research question. Let us assume the user's research question is, "How does chromosomal instability drive tumor progression?" We selected this question from recently published materials on topical examples within the health domain, using "The 150 most important questions in cancer research and clinical oncology series", published in 2017 by the *Chinese Journal of Cancer* [5].

5.2. Ontology File and Document Set

ONTSI requires the user to upload an ontology file and a document set. We used the Human Phenotype Ontology (HPO) in the usage scenario. We selected HPO because of its high complexity resulting from its exhaustive and expert-defined domain coverage of terms and their relationships. HPO is a controlled and standardized vocabulary encoding human disease and phenotypic abnormalities. It also includes annotations in bioinformatics, biochemistry, and human genetics. HPO is an active ontology, consisting not only of over 11,000 terms, but also over 110,000 disease annotations [62]. An example of an HPO term is "blindness," which possesses a superclass of "visual impairment," a subclass of "congenital blindness," and is annotated to be associated with a variety of diseases, such as a variant of colorblindness termed Achromatopsia 2 [63]. Each HPO term describes attributes such as names, conceptual definitions, ontology indexing, term synonyms, class relationships,

logical definitions, and expert commentary, to name a few. For additional details on the Human Phenotype Ontology, see [11,57].

The National Library of Medicine's PubMed is selected as the document set within the usage scenario. PubMed is chosen because of its prominence within the health domain, maintaining more than 30 million citations used within a wide scope of literature and active research endeavors. Data availability limits this usage scenario to a subset of PubMed representing 10,000 document entries. These entries maintain the document title, abstract, and various metadata like authors, published date, and keywords [48].

5.3. Usage Scenario

The user loads ONTSI, finding it in its initial state, as seen in Figure 7.

Figure 7. The initial state of ONTSI.

The user uploads their document set and ontology file by clicking the Upload button, activating the Upload subview, as seen in Figure 8. After confirming a selection, the upload process begins.

Figure 8. Selecting a document set and ontology file for upload.

After the upload process is complete, the user begins typing the research question "How does chromosomal instability drive tumor progression?" into the textbox, finishing with a click of the "Run" button. In response, ONTSI provides the results of its computation, as seen in Figure 9. It presents the first page of 500 pages of documents, which totals 20 document entries. There are percentages to the left of each entry that use text and color to annotate the relevance of each document. This scalar is based on the cluster weightings

within the dimensional space of the document set, where a 100% would be produced by documents within a cluster that aligns with every input feature. The scalar maintains a color scale between red and green, where red is at the zero point and green at 100%. For instance, at the top of the first page there are five documents that present an orange 45.25% relevance rating. Looking at these documents, the user scans the titles of the documents, where some have terms within their titles that relate to the research question.

Figure 9. The results of a search task using the query, "How does chromosomal instability drive tumor progression?"

Some documents at the top of the results could align with the user's research question. To explore further, the user selects a few of the top documents, generating additional document information for inspection, as seen in Figure 10. Doing so, the user encounters a summarized version of their selected documents, which provides metadata and abstracts annotated with words and phrases related to the research question.

The user estimates that these top documents may align with their research question. Therefore, they click on the green "pin" button found at the rightmost point of each document to save their reference for future retrieval from the document set. These references are accessible by clicking the green "pin" button found at the top right of ONTSI to open the Saved List subview.

Although the user has now encountered some documents relevant to their research question, they choose to continue searching. This time, the user decides to take advantage of mediation opportunities when building their query. After closing the Saved List subview,

the user begins a new search. After assessing the important words in their research question, the user types in the term "chromosomal". At the point that they have typed "chromo," they are presented with mediation opportunities, as seen in Figure 11. They inspect these mediation opportunities and add phenotypic terms that align with their research question.

Figure 10. ONTSI after opening the documents "Cancer morphology, carcinogenesis and genetic instability: a background" and "Kaposi's sarcoma–associated herpesvirus–encoded latency-associated nuclear antigen induces chromosomal instability through inhibition of p53 function".

With the aid of mediation, the user builds a three-item query consisting of "abnormal chromosome morphology," "chromosomal instability," and "tumor progression". After asking ONTSI to run with this query, the user encounters a set of results different from the one produced by their earlier search, as seen in Figure 12. Notably, an increased set of 10 documents at a 48.75% rating is encountered. Looking at these documents, the user notices some that are familiar, such as the saved "Genetic instability in human tumors".

Figure 11. ONTSI while the user is presented with mediating opportunities from the expert–defined Human Phenotype Ontology.

From this listing, the user selects two new documents for deeper inspection, as seen in Figure 13. They notice that terms such as "morphology" and "neoplasm" are now being highlighted within the document annotations. The adjusted query based on mediation opportunities has helped promote documents that align with their research question. In this case, the user finds value in the two documents, so they are saved.

Before concluding the search task, the user can upload a different ontology file to investigate how alternate vocabularies may bridge them to their document set. Their encounters could have also allowed them to make the assessment that the document set may not be best to help with their research question. If that is the case, they may upload a different document set, performing another scoping review. In any case, ONTSI provides a search task interface that has been generalized to support plug-and-play capabilities for user-provided ontology files and document sets, allowing users to customize the interface to match their search task objectives.

⬆	abnormal chromosome morphology x	chromosomal instability x		Q Run	🗑			📌	⬇

tumor progression x

Type a term and hit enter to save it.

Front | Previous | Page 1 of 500

(Documents 1-20) | Next | End

48.75% Relevant	Microsatellite **instability** in multiple nonfamilial malignancies.	⌄ 📌
48.75% Relevant	Rayleigh-Taylor **instability** in elastic solids.	⌄ 📌
48.75% Relevant	Cancer **morphology**, carcinogenesis and genetic **instability**: a background.	⌄ 📌
48.75% Relevant	The influence of vertebral **instability** on peridural circulation and concomitant peridural fibrosis formation.	⌄ 📌
48.75% Relevant	Kaposi's sarcoma-associated herpesvirus-encoded latency-associated nuclear antigen induces **chromoso**...	⌄ 📌
48.75% Relevant	STK15 gene overexpression, centrosomal amplification, and **chromosomal instability** in the absence of...	⌄ 📌
48.75% Relevant	Arthroscopic dorsal radiocarpal ligament repair.	⌄ 📌
48.75% Relevant	Cycles of **chromosome instability** are associated with a fragile site and are increased by defects in DN...	⌄ 📌
48.75% Relevant	Genetic **instability** in human **tumors**.	⌄ 📌
48.75% Relevant	Orthostatic **instability** in a population-based study of chronic fatigue syndrome.	⌄ 📌
41.75% Relevant	[Studies of treatment strategy and prognosis on acute myeloid leukemia with **chromosome** 8 and 21 tr...	⌄ 📌
41.75% Relevant	Multicolor fluorescence in situ hybridization (SKY) in mycosis fungoides and Sézary syndrome: search for...	⌄ 📌
41.75% Relevant	Shwachman syndrome as mutator phenotype responsible for myeloid dysplasia/neoplasia through karyot...	⌄ 📌
41.75% Relevant	Robust estimation of experimentwise P values applied to a genome scan of multiple asthma traits identifi...	⌄ 📌
41.75% Relevant	[Research progress in human artificial **chromosome**s(HACs) and the potentials in application].	⌄ 📌
41.75% Relevant	The role of foetal red blood cells in protecting cultured lymphocytes against diepoxybutane-induced **chro**...	⌄ 📌
41.75% Relevant	Y-**chromosome** diversity is inversely associated with language affiliation in paired Austronesian- and Pa...	⌄ 📌
41.75% Relevant	Use of plant genotoxicity bioassay for the evaluation of efficiency of algal biofilters in bioremediation of t...	⌄ 📌
41.75% Relevant	A scan of **chromosome** 10 identifies a novel locus showing strong association with late-onset Alzheimer ...	⌄ 📌
41.75% Relevant	[Clinical and experimental study of 7 cases of acute lymphoblastic leukemia with dic(7;9) (pll;pll)].	⌄ 📌

Front | Previous | Page 1 of 500 (Documents 1-20) | Next | End
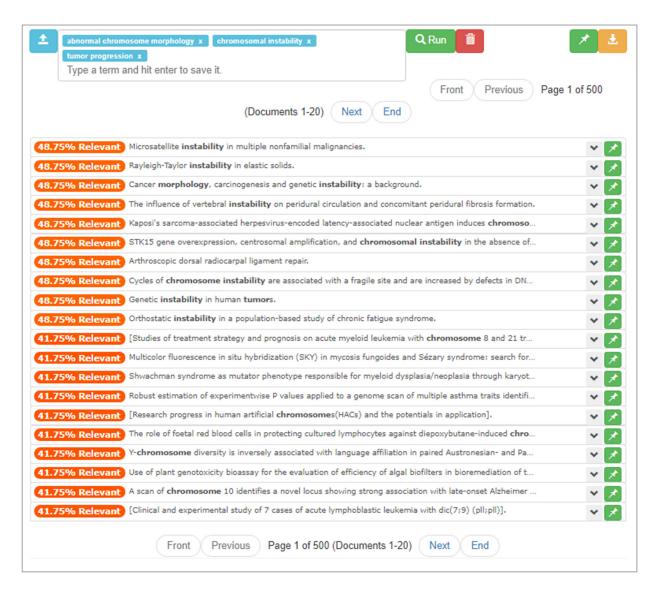
Figure 12. ONTSI after running a new search after the user took advantage of mediation opportunities presented in the generation of the query items.
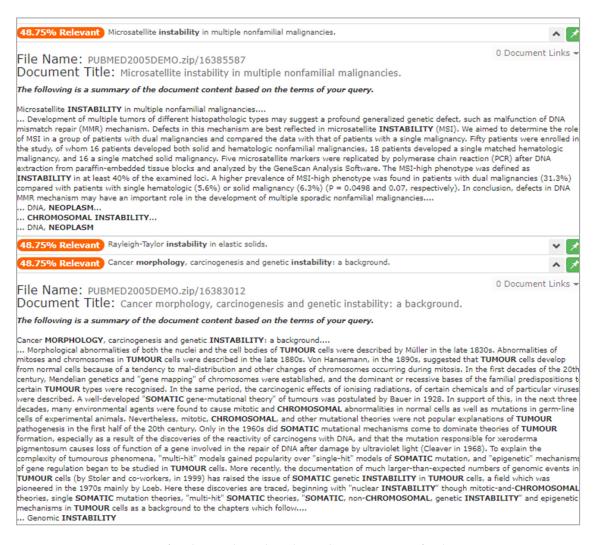
Figure 13. ONTSI after the user has selected new document entries for deeper inspection.

6. Discussion and Conclusions

6.1. Evaluation of ONTSI

We have conducted ongoing, formative, task-driven user evaluations of ONTSI. These evaluations were informally conducted with a few people associated with our research lab; they have provided initial insights into how ontology-supported interfaces for health informatics can support users to perform elaborate search tasks involving large document sets. In these evaluations, we asked the users to perform a targeted set of tasks, such as researching questions outlined in the presented usage scenario. Initial sessions provided general insight into how users search and how ontologies can help mediate such tasks. From these sessions, we have learned a few things, which are itemized below:

- Users are able to quickly transfer their experiences with previous interfaces to use ONTSI (e.g., A, B).
- Users are capable of utilizing ontology files to align their vocabulary with the vocabulary of the domain, even if they are not initially familiar with the ontology's domain or its structure and content (e.g., C, D).
- Users are capable of understanding the requirements of the information-seeking process, expressing their valuation of the support they are provided by the interface as they performed their search tasks (e.g., E, F, G).
- Users felt that mediating ontologies make search tasks more manageable and easier, and not having them would negatively affect their task performance (e.g., E, F, G).

The following are some informal excerpts of some of the comments of those who have used ONTSI:

(A) *"I think with ONTSI, I can immediately it matches my mental models of how I use search interfaces. I type things in, I click run, I go through pages of results".*

(B) *"Once I understood what it was showing me, it helped me. Usually with new tools I tend to read through the documentation or watch videos. And then it still takes me like a while to pick up on them. Like, just running through them and using them a few times. Once you get the hang of it, usually you find success in whatever it's providing you".*

(C) *"But . . . you can get lost in the information too, right? So, if you have like so much so many things related in that ontology, it's like, well, it can be useful. But it could also be a distraction for something that you know. There's this flip side, but I think that's on the searcher to know what they're using and why they're using it. So . . . for me to complete these tasks, if I hadn't had the ontologies listed, then I would have had a much more difficult time. It essentially provided guidance . . . and a structure to something I was unfamiliar with in this case".*

(D) *"I wasn't necessarily intimidated, but I was just like—I don't know what this is. But the background information for the context helped a little bit. A lot of big words, but they did help me when I was looking at the documents that I had to search for to find out which ones I felt best. So even though I did not have full understanding of the words, having them there in that background provided me a kind of help towards finding myself in the space of the question".*

(E) *"I was thinking . . . where (the ontology) would have been helpful. So . . . it would have possibly brought up some of those other terms just from searching a few words and they would be able to make some connections between the text that was provided and some of my search terms (to see) . . . how relevant they were. So, if I was shooting in the dark and hoping for the best, which is what I was kind of doing (without the ontology), at the very least, it would have given you confidence of your actions. Yeah, I think so. A little bit more confidence".*

(F) *"I thought it would be like pretty easy because I (am used to) answering . . . open questions like . . . find the things most relevant. So, this research question is for me . . . just an easier thing to do because I have background in doing that kind of stuff. ONTSI kind of functions like . . . a library tool that is available. This kind of tool felt very familiar to me. I wouldn't say that I'm an expert when it comes to medical knowledge, but . . . I understand . . . basic terminology. So . . . what the terms meant or what they refer to wasn't really . . . an issue. It wasn't really alienating. I have like some general level of confidence just using the terms and trusting the tool as you went along".*

(G) *"Yeah, this (ontology) would have helped because I can find the things that . . . share in common, and that can make it probably much easier to find the relevant documents. Yeah, being able to see the things that certain phrases . . . or words share in common. You can find that common link . . . that can find you the relevant documents".*

In the future, we plan to perform formal, empirical evaluations with users comparing ONTSI to other systems. Such evaluations will help generate new insights into features of interface designs and their qualitative and quantitative measures of how search task performances are affected. Beyond that, such evaluation studies may provide prescriptive guidelines for the design of optimal and effective interfaces r health informatics search tasks. In addition, we intend to further investigate how ontologies and machine learning should be integrated into elaborate and challenging search tasks that need domain-specific knowledge for optimal performance.

6.2. Limitations

The first limitation of ONTSI is the scaling of computational resources. ONTSI in its current state provides a plug-and-play experience that can handle the uploading and

processing of both document sets and ontology files of large sizes. For instance, ONTSI easily handles HPO and its more than 11,000 ontology terms, alongside an extracted subset of PubMed of more than 10,000 documents. Yet, under the load of large-volume document sets and connected suites of ontology files, ONTSI's computational systems may provide reduced responsiveness. To deal with such scenarios, further work is needed to solve overhead limitations—strategies such as pre-hosting common ontology files, establishing API connections to access externally hosted document sets, as well as simply expanding the computational power of our systems.

The second limitation of ONTSI is the support of ontology file formats. ONTSI in its current state can process the core encoded elements within the OWL format, a leading format for encoding ontologies. Yet, the format is quite verbose in its specification, requiring developments beyond the scope of our immediate research objectives. In addition, there are other formats used to encode ontologies that would be valuable to support ontology-supported interfaces for health informatics search tasks.

6.3. Conclusions

In summary, in this paper we began with an examination of the background on the topics of health informatics, machine learning, and ontologies. We then reviewed recent research on health informatics search tasks. Based on this review, we formalized a set of criteria for guiding designers when creating ontology-supported interfaces for health informatics search tasks involving large document sets. We then used these criteria to contrast traditional design strategies for interfaces of search tasks.

To demonstrate the utility of the criteria in the design process, we applied them to structure the creation of ONTSI (ONTology-supported Search Interface), an ontology-supported interface for health informatics search tasks involving large document sets. ONTSI combines five front-end subviews and two back-end computational systems. With these systems, ONTSI supplies a generalized interface that supports users' ability to plug-and-play their provided document sets and an ontology file as a mediating resource within the interface when performing their health informatics search tasks.

The workflow of ONTSI was described and illustrated in a usage scenario. For our scenario, we used the Human Phenotype Ontology to mediate a search task on a subset of the PubMed document set. This usage scenario presented a narrative of a health professional performing a scoping review. Within the scenario, we found that ONTSI allows the user to utilize their ontology resource in a manner that aligns with both the unstructured and structured-like query expansion interface strategy. In the former, the user entered a research question without participating in mediation opportunities. In that case, ONTSI used HPO and WordNet as mediating resources to extend the user's query within an expansion model to generate the results of a search task. In the latter case, the user took advantage of mediation opportunities during their query building. Although this usage scenario provides a single health informatics narrative, we believe value can be generated from both the criteria and ONTSI for health informatics in a broad sense. In this sense, we envision that our efforts can be further expanded to encompass tasks in informatics such as consumer informatics, nursing informatics, and ontology-supported domains beyond health and medicine, to name but a few.

In conclusion, in this paper we generated and proposed a set of criteria that can provide guidance to designers in creating ontology-supported interfaces for health informatics search tasks involving large document sets. We illustrated the utility of these criteria in the context of the creation and demonstration of ONTSI. We provided general insight from ongoing, formative, task-driven user evaluations of ONTSI. We hope to continue this research to promote the design of generalized ontology-supported interfaces for health informatics search tasks involving large document sets.

Author Contributions: Conceptualization, J.D. and K.S.; methodology, J.D. and K.S.; software, J.D.; validation, J.D. and K.S.; formal analysis, J.D. and K.S.; investigation, J.D. and K.S.; resources, Insight Lab, J.D. and K.S.; data curation, J.D.; writing—original draft preparation, J.D.; writing—review

References

1. Wickramasinghe, N. Essential Considerations for Successful Consumer Health Informatics Solutions. *Yearb. Med. Inform.* **2019**, *28*, 158–164. [CrossRef] [PubMed]
2. *The Future of Technology in Health and Health Care: A Primer*; Canadian Medical Association: Ottawa, ON, Canada, 2018.
3. Demiris, G. Consumer Health Informatics: Past, Present, and Future of a Rapidly Evolving Domain. *Yearb. Med. Inform.* **2016**, *25*, S42–S47. [CrossRef]
4. Zuccon, G.; Koopman, B. Integrating Understandability in the Evaluation of Consumer Health Search Engines. *CEUR Workshop Proc.* **2014**, *1276*, 32–35.
5. Gao, C.-F. The 150 Most Important Questions in Cancer Research and Clinical Oncology Series: Questions 67–75. *Chin. J. Cancer* **2017**, *36*, 86.
6. Mehta, N.; Pandit, A. Concurrence of Big Data Analytics and Healthcare: A Systematic Review. *Int. J. Med. Inform.* **2018**, *114*, 57–65. [CrossRef]
7. Thiébaut, R.; Cossin, S. Artificial Intelligence for Surveillance in Public Health. *Yearb. Med. Inform.* **2019**, *28*, 232–234. [CrossRef] [PubMed]
8. Saleemi, M.M.; Rodríguez, N.D.; Lilius, J.; Porres, I. A Framework for Context-aware Applications for Smart Spaces. In *Smart Spaces and Next Generation Wired/Wireless Networking*; Springer: Berlin/Heidelberg, Germany, 2011; Volume 6869 LNCS, pp. 14–25. ISBN 9783642228742.
9. Gibson, C.J.; Dixon, B.E.; Abrams, K. Convergent Evolution of Health Information Management and Health Informatics. *Appl. Clin. Inform.* **2015**, *06*, 163–184. [CrossRef]
10. Fang, R.; Pouyanfar, S.; Yang, Y.; Chen, S.C.; Iyengar, S.S. Computational Health Informatics in the Big Data Age: A Survey. *ACM Comput. Surv.* **2016**, *49*, 1–36. [CrossRef]
11. Köhler, S.; Carmody, L.; Vasilevsky, N.; Jacobsen, J.O.B.; Danis, D.; Gourdine, J.-P.; Gargano, M.; Harris, N.L.; Matentzoglu, N.; McMurry, J.A.; et al. Expansion of the Human Phenotype Ontology (HPO) Knowledge Base and Resources. *Nucleic Acids Res.* **2018**, *47*, D1018–D1027. [CrossRef] [PubMed]
12. Carayon, P.; Hoonakker, P. Human Factors and Usability for Health Information Technology: Old and New Challenges. *Yearb. Med. Inform.* **2019**, *28*, 71–77. [CrossRef]
13. Gamache, R.; Kharrazi, H.; Weiner, J.P. Public and Population Health Informatics: The Bridging of Big Data to Benefit Communities. *Yearb. Med. Inform.* **2018**, *27*, 199–206. [CrossRef] [PubMed]
14. Brewer, L.P.C.; Fortuna, K.L.; Jones, C.; Walker, R.; Hayes, S.N.; Patten, C.A.; Cooper, L.A. Back to the Future: Achieving Health Equity Through Health Informatics and Digital Health. *JMIR mHealth uHealth* **2020**, *8*, 1–16. [CrossRef] [PubMed]
15. Wu, C.M.; Meder, B.; Filimon, F.; Nelson, J.D. Asking Better Questions: How Presentation Formats Influence Information Search. *J. Exp. Psychol. Learn. Mem. Cogn.* **2017**, *43*, 1274–1297. [CrossRef]
16. Talbot, J.; Lee, B.; Kapoor, A.; Tan, D.S. EnsembleMatrix: Interactive Visualization to Support Machine Learning with Multi-ple Classifiers. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Boston, MA, USA, 4–9 April 2009; pp. 1283–1292.
17. Hohman, F.; Kahng, M.; Pienta, R.; Chau, D.H. Visual Analytics in Deep Learning: An Interrogative Survey for the Next Frontiers. *IEEE Trans. Vis. Comput. Graph.* **2018**, *25*, 2674–2693. [CrossRef]
18. Yuan, J.; Chen, C.; Yang, W.; Liu, M.; Xia, J.; Liu, S. A Survey of Visual Analytics Techniques for Machine Learning. *Comput. Vis. Media* **2021**, *7*, 3–36. [CrossRef]
19. Endert, A.; Ribarsky, W.; Turkay, C.; Wong, B.L.W.; Nabney, I.; Blanco, I.D.; Rossi, F. The State of the Art in Integrating Machine Learning into Visual Analytics. *Comput. Graph. Forum* **2017**, *36*, 458–486. [CrossRef]
20. Jusoh, S.; Awajan, A.; Obeid, N. The Use of Ontology in Clinical Information Extraction. *J. Phys. Conf. Ser.* **2020**, *1529*, 052083. [CrossRef]

21. Lytvyn, V.; Dosyn, D.; Vysotska, V.; Hryhorovych, A. Method of Ontology Use in OODA. In Proceedings of the IEEE Third International Conference on Data Stream Mining & Processing, Lviv, Ukraine, 21–25 August 2020; pp. 409–413.

22. Román-Villarán, E.; Pérez-Leon, F.P.; Escobar-Rodriguez, G.A.; Martínez-García, A.; Álvarez-Romero, C.; Parra-Calderóna, C.L. An Ontology-based Personalized Decision Support System for Use in the Complex Chronically Ill Patient. *Stud. Health Technol. Inform.* **2019**, *264*, 758–762. [PubMed]

23. Sacha, D.; Sedlmair, M.; Zhang, L.; Lee, J.A.; Weiskopf, D.; North, S.; Keim, D. Human-centered Machine Learning Through Interactive Visualization: Review and Open Challenges. In Proceedings of the 24th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, Bruges, Belgium, 27–29 April 2016; pp. 641–646.

24. Amershi, S.; Cakmak, M.; Knox, W.B.; Kulesza, T. Power to the People: The Role of Humans in Interactive Machine Learning. *AI Mag.* **2014**, *35*, 105–120. [CrossRef]

25. Zeng, Q.T.; Tse, T. Exploring and Developing Consumer Health Vocabularies. *J. Am. Med. Informat. Assoc.* **2006**, *13*, 24–29. [CrossRef] [PubMed]

26. Arp, R.; Smith, B.; Spear, A.D.; American Journal of Sociology. *Building Ontologies with Basic Formal Ontology*; MIT Press: Cambridge, MA, USA, 2015; Volume 53, ISBN 0262527812.

27. Bikakis, N.; Sellis, T. Exploration and Visualization in the Web of Big Linked Data: A Survey of the State of the Art. In Proceedings of the EDBT/ICDT 2016 Joint Conference, Bordeaux, France, 15 March 2016; Volume 1558.

28. Carpendale, S.; Chen, M.; Evanko, D.; Gehlenborg, N.; Görg, C.; Hunter, L.; Rowland, F.; Storey, M.A.; Strobelt, H. Ontologies in Biological Data Visualization. *IEEE Comput. Graph. Appl.* **2014**, *34*, 8–15. [CrossRef] [PubMed]

29. Dou, D.; Wang, H.; Liu, H. Semantic Data Mining: A Survey of Ontology-based Approaches. In Proceedings of the 2015 IEEE 9th International Conference on Semantic Computing, Anaheim, CA, USA, 7–9 February 2015; pp. 244–251.

30. Livingston, K.M.; Bada, M.; Baumgartner, W.A.; Hunter, L.E. KaBOB: Ontology-based Semantic Integration of Biomedical Databases. *BMC Bioinformat.* **2015**, *16*, 1–21. [CrossRef] [PubMed]

31. Ebner, M. Linked Data Driven Visual Analytics for Tracking Learners in a PLE. In Proceedings of the 13th E-Learning Conference, Munich, Germany, 1 September 2015; pp. 329–331.

32. Jakus, G.; Milutinovic, V.; Omerović, S.; Tomazic, S. *Concepts, Ontologies, and Knowledge Representation*; Springer: New York, NY, USA, 2013; ISBN 978-1-4614-7821-8.

33. Rector, A.; Schulz, S.; Rodrigues, J.M.; Chute, C.G.; Solbrig, H. On beyond Gruber: "Ontologies" in Today's Biomedical Information Systems and the Limits of OWL. *J. Biomed. Inform. X* **2019**, *2*, 100002. [CrossRef]

34. Tobergte, D.R.; Curtis, S. A Survey on Ontologies for Human Behavior Recognition. *J. Chem. Inf. Model.* **2013**, *53*, 1689–1699.

35. Katifori, A.; Torou, E.; Vassilakis, C.; Lepouras, G.; Halatsis, C. Selected Results of a Comparative Study of Four Ontology Visualization Methods for Information Retrieval Tasks. In Proceedings of the Second International Conference on Research Challenges in Information Science, Marrakech, Morocco, 3–6 June 2008; pp. 133–140.

36. Raghupathi, W.; Raghupathi, V. Big Data Analytics in Healthcare: Promise and Potential. *Health Inf. Sci. Syst.* **2014**, *2*, 3. [CrossRef]

37. Russell-Rose, T.; Chamberlain, J.; Azzopardi, L. Information Retrieval in the Workplace: A Comparison of Professional Search Practices. *Inf. Process. Manag.* **2018**, *54*, 1042–1057. [CrossRef]

38. Russell-Rose, T.; Chamberlain, J. Expert Search Strategies: The Information Retrieval Practices of Healthcare Information Professionals. *JMIR Med. Inform.* **2017**, *5*, e33. [CrossRef]

39. Huurdeman, H.C. Dynamic Compositions: Recombining Search User Interface Features for Supporting Complex Work Tasks. *CEUR Workshop Proc.* **2017**, *1798*, 22–25.

40. Zahabi, M.; Kaber, D.B.; Swangnetr, M. Usability and Safety in Electronic Medical Records Interface Design: A Review of Recent Literature and Guideline Formulation. *Hum. Factors* **2015**, *57*, 805–834. [CrossRef]

41. Dudley, J.; Kristensson, P. A Review of User Interface Design for Interactive Machine Learning. *ACM Trans. Interact. Intell. Syst.* **2018**, *1*, 1–37. [CrossRef]

42. Harvey, M.; Hauff, C.; Elsweiler, D. Learning by Example: Training Users with High-quality Query Suggestions. In Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, New York, NY, USA, 9–13 August 2015; pp. 133–142.

43. Soldaini, L.; Yates, A.; Yom-Tov, E.; Frieder, O.; Goharian, N. Enhancing Web Search in the Medical Domain via Query Clarification. *Inf. Retr. Boston.* **2016**, *19*, 149–173. [CrossRef]

44. Anderson, J.D.; Wischgoll, T. Visualization of Search Results of Large Document Sets. *Electron. Imaging* **2020**, *2020*, 388-1–388-7. [CrossRef]

45. Soldaini, L.; Cohan, A.; Yates, A.; Goharian, N.; Frieder, O. Retrieving Medical Literature for Clinical Decision Support BT. *Adv. Soc. Comput.* **2015**, *9022*, 538–549.

46. Wilson, M.L. *Search User Interface Design. Synthesis Lectures on Information Concepts, Retrieval, and Services*; Morgan & Claypool Publishers: Williston, VT, USA, 2011; Volume 3, ISBN 9781608456895.

47. Zielstorff, R.D. Controlled Vocabularies for Consumer Health. *J. Biomed. Inform.* **2003**, *36*, 326–333. [CrossRef] [PubMed]

48. PubMed. Available online: https://pubmed.ncbi.nlm.nih.gov/ (accessed on 18 January 2021).

49. McCray, A.T.; Tse, T. Understanding Search Failures in Consumer Health Information Systems. *AMIA Annu. Symp. Proc.* **2003**, 430–434.

50. Keselman, A.; Browne, A.C.; Kaufman, D.R. Consumer Health Information Seeking as Hypothesis Testing. *J. Am. Med. Informat. Assoc.* **2008**, *15*, 484–495. [CrossRef] [PubMed]

51. Luo, J.; Wu, M.; Gopukumar, D.; Zhao, Y. Big Data Application in Biomedical Research and Health Care: A Literature Review. *Biomed. Inform. Insights* **2016**, *8*, BII.S31559. [CrossRef] [PubMed]

52. Qvarfordt, P.; Golovchinsky, G.; Dunnigan, T.; Agapie, E. Looking Ahead: Query Preview in Exploratory Search. In Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, New York, NY, USA, 28 July–1 August 2013; pp. 243–252.

53. Zuccon, G.; Koopman, B. Choices in Knowledge-base Retrieval for Consumer Health Search. In Proceedings of the European Conference on Information Retrieval, Grenoble, France, 25–29 March 2018; Volume 10772, pp. 72–85.

54. Azad, H.K.; Deepak, A. A New Approach for Query Expansion using Wikipedia and WordNet. *Inf. Sci.* **2019**, *492*, 147–163. [CrossRef]

55. Jimmy, J.; Zuccon, G.; Palotti, J.; Goeuriot, L.; Kelly, L. Overview of the CLEF 2018 Consumer Health Search Task. *CEUR Workshop Proc.* **2018**, *2125*.

56. Capuano, N.; Longhi, A.; Salerno, S.; Toti, D. Ontology-driven Generation of Training Paths in the Legal Domain. *Int. J. Emerg. Technol. Learn.* **2015**, *10*, 14–22. [CrossRef]

57. The Human Phenotype Ontology. Available online: https://hpo.jax.org/app/ (accessed on 10 May 2020).

58. Lüke, T.; Schaer, P.; Mayr, P. Improving Retrieval Results with Discipline-Specific Query Expansion. In *International Conference on Theory and Practice of Digital Libraries*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 408–413.

59. Jay, C.; Harper, S.; Dunlop, I.; Smith, S.; Sufi, S.; Goble, C.; Buchan, I. Natural Language Search Interfaces: Health Data Needs Single-Field Variable Search. *J. Med. Internet Res.* **2016**, *18*, e13. [CrossRef] [PubMed]

60. Solr Cloud. Available online: https://lucene.apache.org/solr/ (accessed on 18 January 2021).

61. Bostock, M. D3.js Data-Driven Documents. Available online: https://d3js.org/ (accessed on 18 January 2021).

62. Kohler, S.; Doelken, S.C.; Mungall, C.J.; Bauer, S.; Firth, H.V.; Bailleul-Forestier, I.; Black, G.C.M.; Brown, D.L.; Brudno, M.; Campbell, J.; et al. The Human Phenotype Ontology Project: Linking Molecular Biology and Disease through Phenotype Data. *Nucleic Acids Res.* **2014**, *42*, 966–974. [CrossRef] [PubMed]

63. Köhler, S.; Robinson, P. HPO Web Browser: Blindness Infopage. Available online: http://compbio.charite.de/hpoweb/showterm?id=HP:0000618 (accessed on 18 January 2021).