DocFlow: A Visual Analytics System for Question-Based Document Retrieval and Categorization

Rui Qiu, Yamei Tu, Yu-Shuen Wang, Po-Yin Yen, and Han-Wei Shen

Abstract—A systematic review (SR) is essential with up-to-date research evidence to support clinical decisions and practices. However, the growing literature volume makes it challenging for SR reviewers and clinicians to discover useful information efficiently. Many human-in-the-loop information retrieval approaches (HIR) have been proposed to rank documents semantically similar to users' queries and provide interactive visualizations to facilitate document retrieval. Given that the queries are mainly composed of keywords and keyphrases retrieving documents that are semantically similar to a query does not necessarily respond to the clinician's need. Clinicians still have to review many documents to find the solution. The problem motivates us to develop a visual analytics system, DocFlow, to facilitate information-seeking. One of the features of our DocFlow is accepting natural language questions. The detailed description enables retrieving documents that can answer users' questions. Additionally, clinicians often categorize documents based on their backgrounds and with different purposes (e.g., populations, treatments). Since the criteria are unknown and cannot be predefined in advance, existing methods can only achieve categorization by considering the entire information in documents. In contrast, by locating answers in each document, our DocFlow can intelligently categorize documents based on users' questions. The second feature of our DocFlow is a flexible interface where users can arrange a sequence of questions to customize their rules for document retrieval and categorization. The two features of this visual analytics system support a flexible information-seeking process. The case studies and the feedback from domain experts demonstrate the usefulness and effectiveness of our DocFlow.

Index Terms—Biomedical systematic review, evidence-based-practice, human-in-the-loop information retrieval, question-based document categorization, question-based document retrieval

1 Introduction

Nowadays, clinical decisions rely on systematic reviews (SR) with up-to-date research evidence to support evidence-based practice (EBP) [1]. Human-in-the-loop Information Retrieval has been applied in the biomedical domain to assist in the SR process, where query algorithms and interactive retrieval systems are used to accelerate SR productions [2]. With the growing size of the literature, efficient and intuitive HIR systems are very much in need to help with relevant biomedical documents retrieval (e.g., published studies or clinical trials) that have high-quality research findings to answer a clinical question (query), thus guiding patient care and inform clinical decisions.

- *Rui Qiu, Yamei Tu, and Han-Wei Shen are with the Department of Computer Science and Engineering, The Ohio State University, Columbus, OH 43210-1132 USA. E-mail: qiu.580@buckeyemail.osu.edu, {tu.253, shen.94}@osu.edu.*
- *Yu-Shuen Wang is with the Department of Computer Science, National Yang Ming Chiao Tung University, HsinChu 300, Taiwan. E-mail: yushuen@cs.nctu.edu.tw.*
- *Po-Yin Yen is with the Institute for Informatics, Washington University School of Medicine, St. Louis, MO 63110 USA. E-mail: yenp@wustl.edu.*

Manuscript received 14 December 2021; revised 23 October 2022; accepted 1 November 2022. Date of publication 4 November 2022; date of current version 2 January 2024.
(Corresponding author: Rui Qiu.)
Recommended for acceptance by J. Kohlhammer.
This article has supplementary downloadable material available at https://doi.org/10.1109/TVCG.2022.3219762, provided by the authors.
Digital Object Identifier no. 10.1109/TVCG.2022.3219762

Many retrieval algorithms and visual analytic systems have been proposed to support biomedical systematic review. The algorithms were developed based on lexical features (TF-IDF, BM25 [3]), semantic features (Doc2Vec [4]), or paragraph to vector transformations [5]. They embed the text's lexical information or hidden semantics into vector representations with a specific dimension and achieve document retrieval by computing vector distances. Because the similarity of documents becomes measurable, many visual analytics systems present a high-level overview of the corpus data to help users explore documents of interest. Despite their usage in various systematic review tasks, existing retrieval methods and visual analytic systems have become gradually untenable due to three major challenges: First, existing retrieval algorithms can only recommend documents semantically similar to users' queries because queries are mainly composed of keywords and key phrases. These documents may not answer clinicians' questions. Consequently, clinicians still have to take a long time to review many irrelevant documents when they attempt to find a treatment for their patients. Second, during the systematic review, clinicians often classify documents according to different perspectives, such as populations and treatments. Without understanding the needs, existing methods can only cluster documents based on the similarity of the entire text, although certain details are considerably different. Finally, as a document retrieval process often involves an iterative refinement of search strategies, clinicians can quickly lose track of how the resulting documents are retrieved during a multi-step query and exploration.

To address these challenges, we introduce DocFlow, a component-based visual analytics system that supports document retrieval and categorization using natural language questions. Each component represents a semantic rule that helps users organize documents. Unlike traditional methods based on similarity comparison, DocFlow retrieves documents that can answer users' questions. To achieve this, we first apply a language model, BioBERT [6], to extract the representations of each document and users' questions. We then train an alignment network on the question-answering (QA) datasets to co-locate the representations of corresponding documents and questions. Accordingly, given a question, the alignment network will distill the document representations by the question before the similarity measurement for retrieval. Thanks to the detailed description of natural languages, DocFlow also supports question-based document categorization. Specifically, it retrieves documents that can answer the question, then applies the BioBERT-QA to locate answers (i.e., phrases) in the retrieved documents. Afterward, the answers are used for categorization. Besides intelligent document retrieval and categorization, DocFlow provides an intuitive visualization interface. Users can arrange a sequence of questions to customize their own rules for systematic review and facilitate information-seeking. We also visualize the retrieved and categorized documents using a Sankey diagram to help clinicians track their information-seeking results.

We demonstrate the effectiveness of DocFlow through quantitative evaluations, case studies, and systematic reviews of COVID-19. The main contributions of our work are as follows:

- *Question-based Document Retrieval.* We present a method to retrieve documents based on natural language questions. Compared to previous works that rely on keywords and key phrases, understanding the semantics of questions can retrieve documents fulfilling users' expectations.
- *Question-based Document Categorization.* We introduce a method to categorize documents based on the answers to users' questions. The categorization considers users' inputs and allows them to organize documents according to various backgrounds and purposes.
- *Visual Analytics System.* DocFlow is a component-based interactive system. Users can flexibly assemble the components with semantic document filtering and categorization functions to seek and analyze information in a large corpus.

2 Related Work

2.1 Human-in-The-Loop Literature Review

Human-in-the-loop solutions that incorporate efficient retrieval methods and interactive visualizations have been shown to be effective in the literature review and systematic review [7]. Existing studies supporting the reviews can be generally categorized into two groups: the works focused on generating static overviews of the corpus and the works focused on improving the entire literature review workflow. Our work is related to both groups.

In the first group, the works of Kumu[1] and [8], [9] support literature exploration by providing an overview of reference and citation relationships. Specifically, two commonly used tools in biomedical SR, VOSViewer [9], and Kumu[1], cluster and visualize the relationships between documents using a citation-based network. CiteRiver [10] extends the citation network by linking topics and venue citations and facilitating their combined analysis. Recently, Bridger [11] has been proposed to facilitate the discovery of valuable scholars and works by generating semantic representations of the authorship. Besides the relationships, many studies generate an overview of a corpus from the perspectives of keywords and topics. KeyVis [12] aggregates the keywords of each paper and recommends related keywords based on users' input. Sci2Tool [13], EvidenceSet [14], and Coremine Medical[2] model the inter-relationship between documents from the topic perspective. The methods facilitate the systematic review by conducting co-word and topical clustering analysis. CREC [15] provides an automatic summarization and highlights medical context terms to help users understand a corpus and term-based query. Recently, studies have visualized the distribution of document embeddings to present the corpus overview. For instance, TRIVIR [16] and Vitality [17] embed text with modern language models and promote the discovery of relevant papers by finding similar representations in the embedding space. More recently, DRIFT [18] combines the embedding-based document projection with an image retrieval visualization to further users' understanding and retrieval performance. Among the related works, two types of visualization choices are commonly picked, (1) node-link diagrams to highlight documents' inter-relationship [19], [20] and (2) 2D scatter plots to display the distribution of the corpus [21], [22]. The studies mentioned above are oblivious to users' needs and can only provide a static view of the corpus. In contrast, DocFlow takes users' questions as input and organizes documents based on the answers to the questions. Its intelligence and flexibility can greatly save burdens on users when reviewing the literature.

In the second group, KGen [23] and Nested-Knowledge[3] streamline the systematic review process into an interactive knowledge extraction, recognition, and retrieval. They provide a series of views and tools to visualize the results in an interactive multi-faceted system and enable users to modify the retrieval interactively. PaperQuest [24] allows users to input papers and recommends relevant papers based on the reference relationships. VisualBib [25] facilitates the creation and review of the bibliography with a series of interactive analysis panels. LitSense [24] and Papers101 [26] are the literature review systems that are most similar to ours. They support users in discovering papers, organizing documents from different sources, and making sense of document discovery through multiple visualization panels. Although effective, the existing works are developed with a fixed, multi-faceted system, and users cannot customize their pipeline when organizing documents. DocFlow allows users to freely assemble intelligent retrieval and categorization components when

1. http://kumu.io
2. https://www.coremine.com/medical/tools.html
3. http://gts.sourceforge.net/

executing a systematic review. It also visualizes the resulting documents at each step to help users track the workflow easily.

2.2 Document Retrieval

Retrieving the documents of interest from a large corpus is critical in a systematic review. Conventional methods such as TF-IDF and BM25 [3] represent a text based on the frequency of each term and compare the lexical information to achieve the goal. Given that the frequency of terms cannot well encode high-level semantics, recently, many works have adopted recurrent neural networks to encode documents and queries [27], [28]. They consider both lexical vectors and text to generate the representations. Afterward, several attention-based language models, such as BERT, ALBERT, or RoBERTA, which were pre-trained on large datasets in a self-supervised manner, have demonstrated their effectiveness in text generation. These language models encoded text into semantic representations for document retrieval. Specifically, DeepCT [29] applies the BERT model to learn the relevance of each term in a document to users' queries and average the relevance scores to determine whether to retrieve the document. In addition, DSSM [30], CLSM [31], and DESM [32] encode the query and the document using their n-gram features or word embeddings independently. Then, they measure the distance between the document and query representations when retrieving documents. We point out that measuring the similarity of a document and a query is insufficient for systematic reviews in the biomedical domain because a document often contains information related to multiple perspectives, such as the treatment of disease and the recovery rates in terms of age. The similarity measurement considers the whole document during retrieval, yet users may be interested in only a part of the text. In contrast, DocFlow retrieves documents that can answer users' questions to prevent such a problem.

3 Background

3.1 Systematic Review

The systematic review of literature is a process for identifying empirical evidence with pre-defined criteria in order to answer a specific clinical question [33]. A typical systematic review usually has (1) a clear set of objectives with predefined eligibility criteria for included studies; (2) comprehensive and reproducible search strategies to identify as many relevant studies as possible; (3) a quality assessment of the research findings; and (4) a systematic synthesis of the study characteristics and findings [34].

In the biomedical and clinical domains, systematic reviews are commonly conducted by domain experts who can clearly define the clinical questions and collaborate with clinical librarians. The experts generate search strategies to obtain a large document corpus from related databases (e.g., PubMed MEDLINE) and execute a document screening process to identify relevant documents that can answer a specific clinical question. However, conducting a systematic review can be labor-intensive and time-consuming [35] and requires months of effort to complete. Specifically, a typical exhaustive search would yield 300 to more than 10,000 citations as the search results for an initial review [36], [37]. Clinicians or systematic review experts must first classify citations as relevant (= included) or irrelevant (= excluded) with a title and abstract level screening. After the triage process, they process the included articles with a full-text level screening. In most systematic reviews, approximately 20–30% of citations are included at the title and abstract level triage, and 1.6–27% are included at the full-text level [36]. In other words, clinicians and systematic review experts spend most of their effort excluding irrelevant or low-quality studies. As the citation screening process (document classification) is one of the most resource and time-intensive steps, such a workload can limit the systematic review results and reduce clinicians' efforts to utilize new research evidence in their practice.

3.2 BERT and Domain-Specific BERT

Bidirectional Encoder Representations from Transformers (*BERT*) is a transformer-based language representation model [38] that obtains state-of-the-art results on a variety of natural language processing tasks. It alleviates the left-to-right unidirectional language constraints utilized by previous SOTA models such as ELMo [39] and OpenAI GPT [40] and introduces a masked language model (MLM) pretraining objective to ensure the representation of each token can capture both the left and the right context. The structure of the BERT model is a multi-layer bidirectional transformer encoder, and each layer relies on a multi-head attention mechanism to compute the contextualized embedding for each token inside an input sequence. Each layer's input is the output of its previous layer. Given an input sequence with n tokens and its embeddings at layer l, the contextualized embeddings of the sequence in layer $l+1$ are computed by (1) performing self-attention on each head and (2) deriving a new embedding and (3) concatenating the embeddings from all heads into a single embedding.

BERT is pretrained to learn contextualized embeddings by solving two pretraining tasks. *First*, *masked language modeling* is a token prediction task by first randomly selecting and replacing tokens in an input sequence with a reserved "[MASK]" token and training the model to retrieve back the original tokens. *Second*, the *next sentence prediction* task aims to distinguish whether a sentence pair appears in a way that the second one succeeds the first one in the original corpus. This sentence-level classification task is performed by projecting a reserved "[CLS]" token's embedding at the very last layer to a set of classification logits for prediction. After the pretraining, down-streaming NLP tasks such as sentiment analysis and question answering can be trained fast by fine-tuning specific task-related datasets.

The original BERT model is pretrained on the concatenation of two huge corpora, BookCorpus, and English Wikipedia to achieve a more general understanding of natural language. More recently, to increase BERT's performance on many domain-specific natural language tasks, many domain-specific BERT variants, which are pretrained on large-scale domain corpora have been proposed. For instance, in the biomedical domain, BioBERT [6] is pretrained on large-scale biomedical corpora and has largely outperformed original BERT in a variety of biomedical text

mining tasks; while in the scientific area, sciBERT [41] is pretrained on large-scale scientific literature, and demonstrates significant performance improvement on five core scientific NLP tasks.

4 Requirement Study

Human-in-the-loop information retrieval (HIR) system in the biomedical domain serves clinicians and biomedical researchers to navigate and optimize their information-seeking process, thus supporting EBP. In this section, we first introduce our formative study of collecting various domain-specific requirements. Then, we analyze these requirements in various HIR tasks and present our summarized task requirements and system requirements for an efficient HIR solution.

4.1 Formative Study

We conducted a formative study with four participants to better understand the domain requirements and propose efficient solutions. Two (E1 and E2) are domain experts with expertise in biomedical informatics and visual analytics, respectively, and the other two are Ph.D. students (E3 and E4, avg. 2.5 year). Specifically, E1 has a clinical background with extensive SR experiences in the biomedical domain. E2 has a computer science background, focusing on information visualization and retrieval. E3's research is related to biomedical informatics, and E4 works on information retrieval and visual analytics. All participants have cultivated (i.e., designed and developed) IR applications, especially for SR.

The study session with participants lasts for approximately one hour. In the session, participants were first asked to describe their current workflow for performing systematic reviews, and their experience with existing HIR systems. Then based on their experience, participants are further asked to list the most useful functions of these systems, their limitations, and the desired functionality for a practical SR application. After the study session, we held multiple discussion sessions with the domain experts to understand the difficulties of knowledge discovery in large scale corpus, identify the potential challenges in the biomedical information-seeking process, and summarize the tasks related to biomedical information retrieval.

4.2 Requirement Analysis

To support an efficient systematic review process in biomedical literature, we have derived a set of requirements from the formative study as follows:

R1: Retrieve documents with a single or a sequence of queries. Given the growing number of research articles, retrieving documents to meet customized needs is critical and necessary. E1 and E3 indicated that two types of query formats are commonly used in the SR.

R1.1: Query with question-format phrases or keywords. Clinicians and biomedical researchers often format their queries as concise question-format phrases for a quick exploration, e.g., treatment of ADHD.

R1.2: Query with natural language questions. Querying with natural language questions can avoid retrieving inaccurate documents that are relevant but do not contain answers to the questions. For instance, E1 wants to find out potential solutions to COVID and inputs a question: "How to prevent secondary transmission of COVID in a community setting?" The question provides enough context to filter many documents which explain the mechanism of COVID transmission and locate the papers that cover the possible prevention of COVID.

R2. Categorize documents based on user-desired perspectives. Systematic reviews usually involve categorizing documents for different purposes. For example, it is common to categorize biomedical research studies based on the population, the type of treatment, and the setup of control groups. In a systematic review, information that interests users should be recognized from documents and used to categorize documents for further analysis. E1 has mentioned that the existing clustering methods, especially data-driven topic modeling, cannot separate documents well from user-desired perspectives. For instance, data-driven topic modeling may cluster COVID-19 treatment studies based on various treatments, while clinicians may be more interested in understanding treatments based on different comorbidity.

R3. Facilitate the information-seeking process through effective visualization. All experts expressed the need for effective visualizations and interactions to facilitate exploration and keep track of the document retrieval process. The requirements can be further decomposed into three perspectives.

R3.1 Keep track of the document retrieval process. A typical SR process involves a series of document filtering and categorization operations. However, all of our domain experts indicated that existing visual analytic systems with static multifaceted layouts are not tailored to create, explicitly display or keep track of the customized retrieval pipeline in a visually friendly way. Also, the process of designing and establishing a retrieval pipeline should be an interactive process, which involves many trials, explorations, and modifications. Thus, an effective visualization system should be able to not only visualize the multi-step retrieval pipeline but also enable users to manipulate and fine-tune the entire process after its construction interactively.

R3.2 Identify similar documents. Estimating the similarity between documents can help clinicians find relevant studies and discover new research directions. E2 and E4 stated that an effective visualization could help users quickly figure out which documents are similar, which are dissimilar, and how many categories are there in a corpus.

R3.3 Perceive the results categorized from multiple perspectives. E2 and E4 further added that documents could be categorized according to different purposes, and it will be good to understand the relationships between the documents in different categorizations. For instance, one can categorize a clinical trial by the studied population and tested treatment. A multi-perspective visualization of these documents can help users better understand the relationships between populations and treatments.

5 Approach

Before we introduce our algorithm and visualization system in detail, we first describe our research components and how they work together to solve the requirements mentioned above.

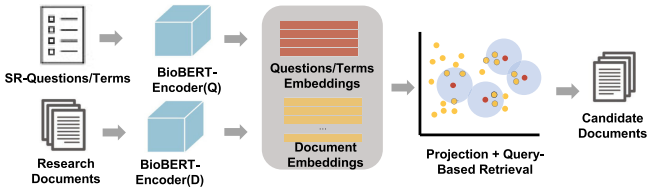


Fig. 1. Illustration of the question-based document retrieval.

5.1 Approach Overview

Question-Based Document Retrieval. We present a method to retrieve documents that can answer users' questions (*R1*). To achieve it, we design a dual-BioBERT language model to embed corresponding questions and documents in the same space. Accordingly, given a question, the documents where the embeddings are close to that of the question will be retrieved.

Question-Based Categorization. Our method categorizes documents based on the answers to questions (*R2*). Specifically, we train a question-answering model to extract a sequence of words and phrases from the documents that can answer the given questions. The words and phrases are then used for categorization. Users also can categorize documents hierarchically from different perspectives by asking a sequence of questions.

Visual Analytical System: DocFlow. We provide a component-based visual analytic system to help users track their document retrieval and categorization process (*R3*). It supports flexible *drag-and-drop* interactions for users to seek information and convey the relationships of documents when the documents are hierarchically categorized from multiple questions.

5.2 Query-Based Document Retrieval

5.2.1 Overview of the Dual-BioBERT

BERT has been proven to capture the semantics of natural languages. By leveraging its variant, BioBERT, which is trained on the biomedical corpus, we design a dual-BioBERT to embed the documents and questions for document retrieval. It consists of two BioBERT encoders, $E_D(\cdot)$ and $E_Q(\cdot)$, that encode documents and questions into d-dimensional vectors, respectively (d is 768).

As illustrated in Fig. 1, the algorithm consists of four stages: (1) *Document Encoding Stage:* $E_D(\cdot)$ projects all documents into vectors $e_d \in R^d$, which is untouched once generated. (2) *Question Encoding Stage:* Given a question q, $E_Q(\cdot)$ encodes the question into a vector $e_q \in R^d$. (3) *Embedding Alignment:* Users may be interested in only a certain part of a document when seeking information. In other words, the embedding e_d, which represents the whole information of a document, should be updated according to the question. To implement this idea, we incorporate an attention layer on top of the dual-BioBERT. It transforms the concatenation of e_d and e_q (i.e., a 1536D vector) into e'_d (i.e., a 768D vector). We then use the cosine similarity to measure the distance between the question e_q and the question-related document e'_d. (4) *Document Retrieval:* We retrieve documents based on the distance between each pair of e_q and e'_d. Note that the transformation is achieved using only one attention layer. The computation cost is inexpensive, although the complexity of the alignment is linear to the number of documents. We also apply the FAISS [42] to boost the performance when retrieving documents from a large collection.

5.2.2 Training of the Dual-BioBERT

Our Dual-BioBERT encodes documents and questions into semantic embeddings, followed by distilling the document embeddings based on the given question. The distillation enables the documents and the question to be considered in the same space, such that the retrieved documents can answer the user's question. Accordingly, we exploit contrastive learning to train the network [43], in which the task is to distinguish whether a document and a question are related. Specifically, let $P = (q, p^+, p_1^-, p_2^-, \ldots, p_n^-)$ be a training sample, which contains a question q, a positive document p^+ that includes the answer to question q, along with n negative documents $(p_1^-, p_2^-, \ldots, p_n^-)$ that are randomly selected from the question-answering dataset (details are in the 5.2.3). The InfoNCE loss [43] is adopted to train the network

$$L(q, p^+, \quad p_1^-, \ldots, p_n^-) = -log \frac{e^{sim(q,p^+)}}{e^{sim(q,p^+)} + \sum_{i=1}^{n} e^{sim(q,p_i^-)}}. \tag{1}$$

Typically, the loss aims to co-locate the questions q and the positive documents p^+ while pushing the question q and the negative documents p^-. Given the summation of $e^{sim(q,p_i^-)}$ is over multiple negative documents, the loss is less sensitive to noise since p^- is randomly selected from datasets without labor-intensive labeling. All parameters in the Dual-BioBERT are updated when training.

5.2.3 Training Data

We construct training samples *(query, positive docs, negative docs)* to achieve two types of questions for interactive document retrieval (*R1.1* and *R1.2*). Details are as follows.

Natural Language Questions. to construct samples of natural language questions, we take advantage of the well-developed question answering datasets: SQuAD [44], TriviaQA [45], Natural Questions [46], and emrQA [47]. All datasets are constructed for extractive QA tasks and are organized in the format of *(question, context, answer)* pair. Based on these well-constructed datasets, we collect positive documents and negative documents for each question by taking each question's original context as the positive example and randomly selecting n other questions' contexts as the negative ones.

Question-Format Phrases. We generate samples for question-format phrases using only the emrQA dataset because domain experts have annotated the symbolic representation of medical events and attributes. For instance, clinical note "nitroglycerin 40 mg daily" is being aligned to $MedicationEvent < Medication = nitroglycerin, dosage = 40mg >$. We extract those attributes and concatenate them into question-formate phrases. Following the previous example, we get *dosage of nitroglycerin?* to represent the question "What is the dosage of nitroglycerin?" We follow this pipeline to generate question-format phrases (phrases, positive documents, negative documents) for training.

5.3 Question-Based Document Categorization

Clinicians must read and summarize relevant documents from multiple perspectives to perform a comprehensive

systematic review (*R2*). Thus, after retrieving relevant documents that can answer a question, it is helpful to categorize these documents based on how they answer the question. To meet such a requirement, we train a question-answering model, BioBERT-QA, to extract the answers from documents to the given question. The answer could be keywords or keyphrases used to represent documents. The embeddings of phrases are then used to categorize the documents.

5.3.1 BioBERT-QA for Answer Extraction

Our BioBERT-QA aims to extract the answer from the document to a question. By defining the answer as a sequence of words, we can train the network to identify only the start and the end tokens. Specifically, the BioBERT-QA has a question-answering layer on top of the BioBERT. It takes the embedding $t_i \in R^d$ of token i generated by the BioBERT as input and classifies the token as a start or an end of the answer. We apply the binary cross-entropy loss to train the network. In our implementation, we train the BioBERT-QA on the biomedical emrQA dataset for 4 epochs with a learning rate of 5e-5 and a batch size of 32. During inference, we compute the probabilities of token i being the start and end of the answer as

$$P_{S_i} = \frac{e^{S \cdot t_i}}{\sum_j e^{S \cdot t_j}} \qquad P_{E_i} = \frac{e^{E \cdot t_i}}{\sum_j e^{E \cdot t_j}}. \tag{2}$$

where S and E are learnable parameters in the network. Accordingly, the sequence of words with the highest score is selected as the answer, which is computed using

$$score = P_{S_i} + P_{E_j}. \tag{3}$$

Algorithm 1. Question-Based Document Categorization

Input: documents group $P : (p_1, p_2, \ldots p_n)$, feature f, feature-based query q_f, similarity threshold: top_k
Output: Feature categories F with each feature's corresponding documents

```
D(d_1, d_2, ...d_m) = question-based retrieval(D, q_f, top_k)
  //Retrieve documents which cover the answer through question-based document retrieval
(f_1, f_2, f_3, ...f_m) = BioBERT-QA (d_1, d_2, ...d_m) //Extract the answer tokens with BioBERT-QA model from each retrieved document.
Extract ([e_f11, e_f12, ...], ...[e_fm1, e_fm2...]) //Extract each answer tokens' embedding from BioBERT-QA
[e_ans1, e_ans2, ...e_ansm] = (avg([e_f11, e_f12, ...]), ...avg([e_fm1, e_fm2 ...])) //Average the embeddings of each answer as the representation of the answer
for k = 2: numOfAnswers do
    centroids, inertia[k], samplesOfEachCluster = Kmeans(k, [e_ans1, e_ans2, ...e_ansm])
    if inertia[k]-inertia[k-1] <= inertia[k-1]-inertia[k-2]
      then
        return k, samplesOfEachCluster //Best number of cluster achieves when the decreasing speed of inertia start to reduce
      else
        continue
      end
end
```

5.3.2 Categorization Based on Answer Embedding

Since the answer often spans a sequence of words, we average the embedding of each word to represent the answer and use the averaged embedding to categorize documents. Specifically, we apply the k-means algorithm to cluster the answer embeddings, where the number of clusters k is a user-defined parameter. To save users' load on tuning this parameter, by default, k is a value determined based on the within-cluster-sum of squared errors. We initialize $k = 2$, steadily increase the value of k, repeat the clustering method, and then apply the Elbow method [48] to pick the best parameter. Algorithm 1 shows the categorization details.

6 Visual Analytics System: DocFlow

DocFlow is a component-based visual analytics system that can facilitate a flexible and interactive information retrieval process. In this section, we first introduce design principles for designing and implementing DocFlow, then illustrate the details of the DocFlow interface, including key components and interactions employed.

6.1 System Design Principles

We designed DocFlow prototype based on (1) the feedback from our formative study, (2) summarized domain requirements (Section 4), and (3) regular meetings and discussions with domain experts (E1 and E2). We formulate the design into three principles:

1) *DP1: Component-based system design.* Considering the complexity and commonality of a typical document retrieval process, DocFlow should allow users to build a customized retrieval pipeline with composable and reusable components. As a systematic review process often consists of multiple filtering and categorization operations, a component-based system can achieve the flexibility of custom operations. In addition, different components should have low coupling in functionality and low redundancy in data usage.
2) *DP2: Interactive visualization system for pipeline construction and tracking.* DocFlow should support easy user interactions for composing a retrieval pipeline with components and links. Visualization methods can be an integral system for textual data analysis. Furthermore, the system should also visualize the entire pipeline to help users keep track of the retrieval history.
3) *DP3: Reproducible and sustainable pipeline.* DocFlow should support saving and loading user-constructed pipelines to reproduce the document retrieval and categorization process. The saved pipeline should contain all the configuration information, including component layout, links, user input on each component, etc.

6.2 Retrieval Pipeline as Node-Link Diagram

We developed DocFlow as a component-based visual analytics system to fulfill *DP1* and *DP2*. DocFlow visualizes a retrieval pipeline with a node-link diagram (Figs. 6*p* and 2).

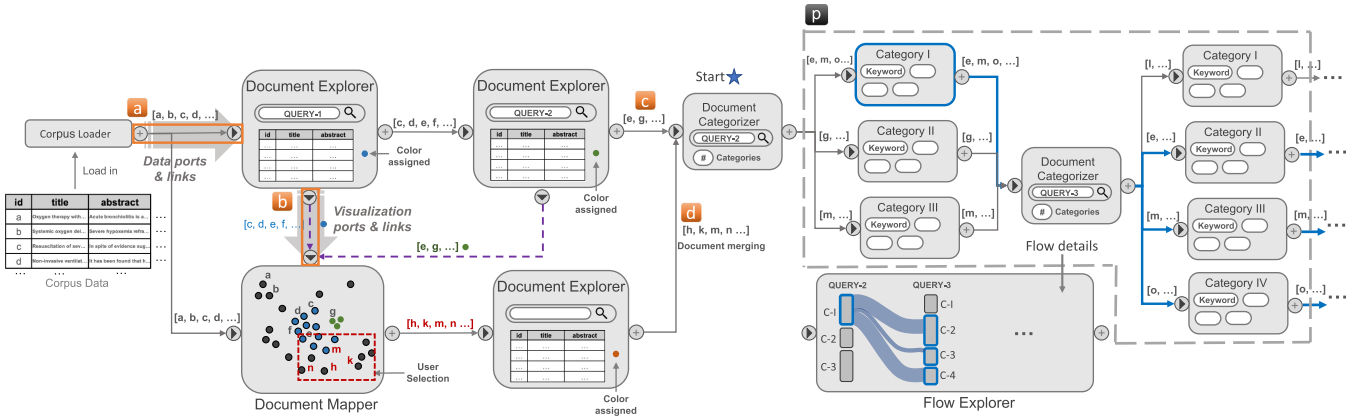


Fig. 2. Illustration of components in DocFlow retrieval pipeline and the flow of documents. Component types are labeled in the pipeline. There are two types of connections between components: (a) data links through data ports; (b) visualization links through visualization ports. The transmission of document sets is denoted by a vector of IDs. The assigned color of each document explorer is shown with a sample color dot and is transmitted to the document mapper as a visual parameter. The flow of documents between categorization components can be zoom-in through a Flow Explorer to check flow details like magnitude.

Each node is a component that performs a pre-defined task, such as loading, embedding, retrieving, etc., and exposes four ports for data receiving and transmitting (Fig. 2). Specifically, ports on a component's left and right sides are data ports, and ports on the top and bottom sides are visualization ports (Fig. 2*a*, *b*). A link connecting different components indicates data transformation from one component to another.

6.2.1 Link and Primitive Elements

DocFlow supports two types of links to connect nodes. Links connecting *data ports* are *data links*, and links connecting *visualization ports* are *visualization links*, as illustrated in Fig. 2*a*, *b*. The transmission of a link can be either a document set or visual parameters. Specifically, a *data link* is in charge of transmitting data links for various retrieval and categorization tasks, while *visualization link* transmits document sets to be highlighted and visual parameters.

Document Set. A document set is a collection of documents that are uploaded, selected, or retrieved from an output component. A document set contains the information to access the corresponding documents, including titles, abstracts, and embeddings. The transmitted documents can be used for further retrieval, categorization, or visualization. In Fig. 2, the transmission of a document set is denoted as a vector of document IDs passed between components.

Visual Parameter. Visual parameters are associated with documents during visualization. In DocFlow, the parameters are the colors of the documents, as illustrated in Fig. 2*b*.

6.2.2 Node Components

DocFlow contains six different node components to fulfill interactive information retrieval. Details are described as follows.

(a) Data Loader. The data loader allows users to upload their own dataset in a tabular format (Fig. 8a, a_1). It is the starting point of a retrieval diagram.

(b) Embedding Generator. The embedding generator contains multiple pre-trained embedding networks for users to select. Users also have to specify the content of interest, such as *abstract*, to generate document embeddings. This component is required when achieving document retrieval and categorization (Fig. 8b, b_1).

(c) Document Explorer. The document explorer is a tabular view to present the documents with details. Users can select a subset of attributes to display during visualization. They also can type a question on the top of the table to retrieve documents. A natural language question or a question-format phrase that ends with a question mark "?" will trigger the retrieving process. The tabular view will update immediately. If the phrase does not end with a question mark, the component will execute a lexical search to find items that contain the matched text. Users can further control the number of documents to retrieve. The threshold picker (Fig. 6$c_1 threshold$), a line chart showing the sorted similarity between the question and documents, helps users select the value. It deserves noting that the first document explorer in the pipeline should be linked from a *embedding generator* to receive both the raw data and the generated embeddings to enable question-based retrieval and categorization. Besides, if a document explorer connects to multiple *document explorers* or the resulting categories of a *document categorizer* (Fig. 5p, $c_{3(full)}$ receive data source from both d_{2-3} and d_{2-5}), it merges all documents into a unified document set for further exploration.

(d) Document Categorizer. The document categorizer extracts phrases from documents that can answer the given question and then categorizes documents according to the phrases. In other words, the output of this component is document categories. The label of each category is composed of the top three most frequent tokens in the answers, e.g., "adults, children, older" (Fig. 5d_{1-2}) are the top three most frequent tokens in the category (Fig. 5$d_{1-2full}$). By stacking multiple document categorizers, users are allowed to hierarchically categorize documents and keep track of the flow of documents between categories (see Fig. 5d_1, $d_{1-1full}$). Besides, users can examine the categorization results with a document mapper and fine-tune the number of categories.

(e) Document mapper. The document mapper is a scatter plot that shows the distribution of documents. Each dot

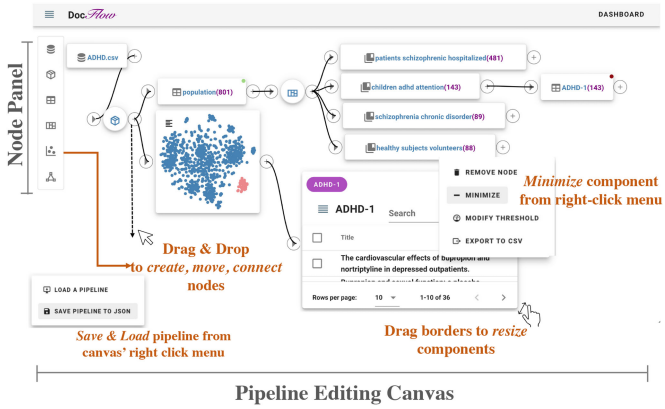


Fig. 3. DocFlow prototype interface. On the left, the node panel lists all supported components, which can be dragged and groped to the pipeline editing canvas to create a pipeline.

represents a document embedding that has been distilled according to the given question. In other words, the scatter plot reveals the document-level similarity in which documents close to each other are more likely to answer similar questions. We apply UMAP [49] to project each document embedding from a $768D$ vector to a $2D$ space because of its efficiency and superior performance on non-linear projection. When a document mapper connects to a document explorer, the document embeddings will be transferred and rendered as scattered points (Fig. $7e_1$, e_2). Colors assigned to the retrieved documents will also be used for highlighting when rendering the dots. In addition, when a document categorizer connects to a document mapper, it receives the embedding of all extracted answers (Figs. $5e_1$ and $6e_1$) for users to study the categorization results and fine-tune the number of categories. A document mapper can receive multiple document sets and visual parameters from different document explorers and highlight different groups of documents. Users are also allowed to directly select dots on a document mapper and check the documents of interest by connecting a document explorer.

(f) Flow Explorer A. The flow explorer is a Sankey diagram [50] that shows the hierarchical categorization results achieved by a series of *document categorizer* (Fig. $2p$, Fig. $8f_1$). Users can specify the start and the end *document categorizers* in a retrieval pipeline to create the *flow explorer*. In the Sankey diagram, each column shows a categorization process, and each rectangle on a column is a category, where its height implies the size of the category. The ribbons connecting consecutive columns reveal the relationships between categories. Users can zoom and select nodes when studying the categorization results. The documents hierarchically categorized will be highlighted.

6.3 User Interaction

DocFlow is equipped with user interactions to support users to quickly build and keep track of their retrieval pipeline (*DP2*). As shown in Fig. 3, users can create an information-seeking process on the Pipeline Editing Canvas. We show the instructions below. (1) Create nodes (DocFlow components), resize, and reposition the nodes in an intuitive drag-and-drop manner. A node panel would guide users on how to create a node. (2) Hold and drag data ports of different nodes to create a link for transmitting a document set or the visualization ports for transmitting the visualization parameters. (3) Resize each node to a small view, in which only the necessary icons, questions, and the number of documents are displayed, to get a clear overview of the pipeline, as shown in Figs. 3, 7, and 5. The document categorizer can further hide all the extracted categories (Fig. $8p_2$) to reduce the space. (4) Save and load the existing pipeline (*DP3*) for resume and results sharing. Through multiple case studies, we demonstrate the effectiveness and usability of these interactions in facilitating the systematic review.

7 Evaluation

In this section, we first evaluate our question-based document retrieval algorithm and DocFlow through quantitative experiments. Then we demonstrate the effectiveness of DocFlow by inviting clinicians to perform case studies.

7.1 Evaluation of Question-Based Document Retrieval

7.1.1 Retrieval Performance

To evaluate our fine-tuned BioBERT-based encoders $E_D(\cdot)$ and $E_Q(\cdot)$, we first measure the accuracy of question-based document retrieval in four QA test datasets. Specifically, our BioBERT-based encoders $E_D(\cdot)$ and $E_Q(\cdot)$ are trained on four question-answering datasets, SQuAD, TriviaQA, NaturalQA, and emrQA to generate similar document embedding and query embedding if the document is the corresponding context of the query. Following the training setting, we conduct and report the accuracy of extracting the correct context from the top 10, 20, and 100 most similar document embeddings based on a query. Results generated by BM25, DSSM, and DESM are also reported for comparison in Table 1. Our BioBERT-based encoder achieves better retrieval accuracy in small retrieval numbers, especially in the emrQA dataset. The accuracy of retrieval in the top-10, 20 are improved by around 20% compared with existing methods. We believe the superior performance in the emrQA dataset is because we use BioBERT as the base encoder. We also notice that the lexical-based method (BM25) performs better than our model in the SQuAD dataset. After checking the details of SQuAD, we consider this is because the dataset was collected from Wikipedia articles on various topics and there is a high lexical overlap between questions and their corresponding context, which makes lexical-based methods like BM25 superior. However, after the ablation study (Section 7.1.2), we found that it is still beneficial to include it in our training dataset.

7.1.2 Ablation Study

We conducted an ablation study on a set of IR tasks to analyze the necessity of training on both biomedical domain QA dataset (emrQA) and the general QA datasets (SQuAD, TriviaQA, Natural Questions). The results are shown in Table 2. We found that although the emrQA dataset is helpful in learning domain-specific knowledge of the biomedical literature, using it only is not sufficient to have a good performance. The result in row_5 indicates that training on the general QA datasets can further improve the model's

TABLE 1
Top 10, Top-20 and Top-100 Retrieval Accuracy on Test Sets, Measured as the Top Number of Retrieved Passages That Contain the Correct Background

Method	Top 10				Top 20				Top 100			
	emrQA*	SQuAD	TriviaQA	NaturalQA	emrQA*	SQuAD	TriviaQA	NaturalQA	emrQA*	SQuAD	TriviaQA	NaturalQA
BM25	53.9	**55.8**	54.2	49.2	57.7	**68.8**	66.9	59.1	63.2	**80.0**	76.7	73.7
DSSM	50.9	52.3	69.6	70.5	60.8	63.2	79.4	78.4	70.6	77.2	85.0	85.4
DESM	55.2	55.6	72.1	73.4	68.8	66.9	80.3	81.9	73.5	74.9	85.3	87.5
QDR(Ours)	**68.9**	54.1	**76.5**	**77.0**	**73.2**	64.4	**82.5**	**83.0**	**80.4**	71.0	**88.7**	**87.7**

retrieval performance. The reason is that some clinical questions are expressed in a general question format, e.g., *"What treatment has the patient had for his ADHD?"*, which can be well learned from the general QA datasets. We also evaluate our embedding alignment network by comparing the retrieval performance using aligned document embeddings (row_1) and pre-compute document embeddings row_5. The result indicates that our aligned document embeddings can improve the retrieval accuracy from 72.5 to 79.7.

7.1.3 Efficiency Analysis

Our model is being tested on Nvidia Tesla V100 GPU for document retrieval and categorization tasks. We report the time cost at each stage in Table 3. The offline and online refers to performing the embedding generation with static document sets and performing document retrieval and categorization tasks with incoming queries, respectively. With the boosting of GPU, question-based document embedding generation can be done in an efficient manner to prepare for the retrieval task. The encoding of the query is generally faster than the encoding of the documents because of the shorter text length. In order to test the efficiency and scalability of retrieval, we construct three test datasets with 1 K, 5 K, and 10 K documents from COVID-19 Open Research Challenge Dataset [55] and the queries in the challenge to perform document retrieval. The result demonstrates that the question-based document retrieval method can handle the retrieval task with a large number of research documents at a faster speed. The results also demonstrate the scalability of our question-based document retrieval and categorization methods.

7.2 Evaluation of DocFlow in SR Tasks

To evaluate the retrieval performance of DocFlow in assisting information retrieval, we used four completed systematic reviews conducted by domain experts (Table 4) as benchmark. Each systematic review has specified the requirements of studies that can be included as relevant. Besides, the search strategy to collect the initial document set and final included documents are also listed. Since many of the requirements are not stated in a question format, we need to modify these requirements into questions or question-format phrases to perform question-based retrieval.

Evaluation Process. To perform a systematic review in DocFlow, the typical process includes (1) identify the key clinical problems and covert them into one or a series of questions, (2) design a sequence of retrieval and categorization as a review pipeline, (3) drag the components to construct the pipeline in DocFlow as a retrieval flow diagram, (4) interactively explore and fine-tune the retrieval process. We compare the final retrieved documents from DocFlow with the documents included in the completed SR.

Performance Measure. To evaluate the performance of DocFlow retrieved documents, we use recall (R) over different percentages of question-based document retrieval. Specifically, each SR is converted into one or multiple queries that fit the requirement of DocFlow and then retrieves the top K percent of documents as the computed result. The queries used for each retrieval are listed in Table 4.

Result. The recall is computed and visualized in Fig. 4. We observed that although the efficiency of retrieving all relevant articles varies across different SRs, they all can achieve satisfying recall at around the top 30% retrieval with DocFlow. For SR1, which requires retrieving 21 relevant documents from 1938 documents, our question-based method can retrieve 18 true positive articles from the top 20% of relevant documents returned by the method. As for SR2, we can retrieve 4 out of 6 studies from the top 25% of articles. In SR3, we retrieved 12 out of 15 included articles from the top 25%. In SR4, 24 out of 32 articles were retrieved from the top 20% with a single retrieval question. On

TABLE 2
Performance of Document Retrieval Task With emrQA Dataset Measured by Top 20 and Top 100 Accuracy Under Different Training and Model Setting

Training/setting	emrQA@ Top20	emrQA@ Top100
w/o embedding concatenation	72.5	82.8
emrQA	61.7	73.2
emrQA+TriviaQA	66.1	77.7
emrQA+TriviaQA+SQuAD	74.9	83.2
emrQA+All*	**79.7**	**86.9**

* *Indicates the training with TriviaQA, SQuAD and NaturalQA dataset*

TABLE 3
Time Cost of Offline and Online Computing in COVID-19 Open Research Document Retrieval Task

Operation	Offline	Online
Per Document $E_D(\cdot)$ encoding	1.1 ms	-
Per Document Answer Extraction	-	3.2 ms
Per Query $E_Q(\cdot)$ encoding	-	0.6ms
Retrieve(1,000)	-	220 ms
Retrieve(5,000)	-	830ms
Retrieve(10,000)	-	1,430ms

Offline operation refers to encoding static documents before the retrieval and categorization task. Online refers to performing retrieval and categorization tasks with incoming queries.

TABLE 4
Description of Four Complete Systematic Reviews Used in the DocFlow Evaluation

System Review	Query strategy in DocFlow	Input/Include
Natural language processing and text mining of symptoms from electronic patient-authored text data [51]	1. What symptoms have been detected with natural language processing or text mining skills? 2. Source of text data?	Input: 1198 articles Included: 21 articles
Automated Deterioration Detection Using Elec- tronic Medical Record Data in Intensive Care Unit Patients [52]	1. How to use electronic medical record data to identify deterioration in intensive care unit patents?	Input: 489 articles Included: 6 articles
Safety and Usability Guidelines of Clinical Infor- mation Systems Integrating Clinical Workflow [53]	1. What clinical information system is being studied? 2. Safety and usability guidelines of clinical information system integrating clinical workflow?	Input: 2241 articles Included: 15 articles
Multiple Myeloma Genomics [54]	1. What genomic variants have been found to be associated with poor prognosis in patients diagnosed with multiple myeloma?	Input: 117 articles Included: 32 articles

average, DocFlow reaches 100% recall at 43% of retrieval across four SRs. Compared with most related work [56] on replicating SRs with information retrieval, which retrieves the top 65.2% of documents on average to get all true positive items, our method improves the performance by 33.8%.

7.3 Case Study

We apply two case studies to illustrate how DocFlow benefits SR. Two biomedical experts, E1 and E3, were invited to participate in the studies. Initially, we arranged a one-hour tutorial to introduce the functionality of DocFlow. During the study, we let the experts freely use DocFlow and assisted them only when they had questions.

7.3.1 Long COVID Discovery

COVID-19 is an ongoing global pandemic that has taken over 6 million people's lives and caused significant social and economic disruption. Although most COVID patients can fully recover, some may experience long-term effects from their infection, known as post-COVID conditions (PCC) or long COVID. Constant updates of knowledge and evidence to support the ongoing fight against this infectious disease are urgently needed. In this study, E1 and E3 were interested in an IR task to keep track of the latest research results and regulations about Long COVID and Post-COVID conditions, and specifically, the population likely to experience long covid and their symptoms. They used DocFlow to retrieve articles from the latest COVID-19 Open Research Dataset (CORD-19), which contains 1,056,660 research articles (documents) on 2022-06-02.

E1 and E3 first filtered the documents based on the publication time (after 2021-01) and the keyword "long covid" before the exploration. 7396 documents were filtered, and they were exported into a CSV file. Each document contains a title and an abstract. E1 loaded the CSV file using the document loader (Fig. $5a_1$) and connected the loader with an embedding generator (Fig. $5b_1$). Then, E1 and E3 began their exploration and attempted to find answers to their questions using DocFlow.

Q1: What populations are likely to have long covid or post-covid? E1 first connected a document explorer with query Q1 to the embedding generator (Fig. $5c_1$) and retrieved the 908 most relevant documents. She found that most retrieved documents describe the population with long covid, including *adult cohorts*, *outpatients* and *children*. E1 opened the Threshold Fine-tuning View (Fig. $5c_{1-threshold}$) of the document explorer and noticed that the filtering threshold was set to 0.09 (i.e., the elbow point of the line). The document with the relevant score of 0.09 was titled "Patient Symptoms in Adult Patients 1 year after Coronavirus Disease 2019: A Prospective Cohort Study," which E1 considered relevant to her interest. E1 then looked at the documents above and below the threshold by hovering on the line chart. She found that the documents with high scores are relevant at the title level. For instance, the paper "Assessment of Post-Covid Symptoms in Covid-19 Recovered Patients" has a relevant score of 0.24. In contrast, documents with low scores likely focus on the impact of long covid and treatments. For instance, a document with a score of 0 entitled "COVID-19 and protected areas: impacts, conflicts, and possible management solutions," describes the social impact of long COVID.

E1 then added a document categorizer (Fig. $5d_1$) at the back of the document explorer (Fig. $5c_1$) and obtained the categorization results of the documents. Specifically, there

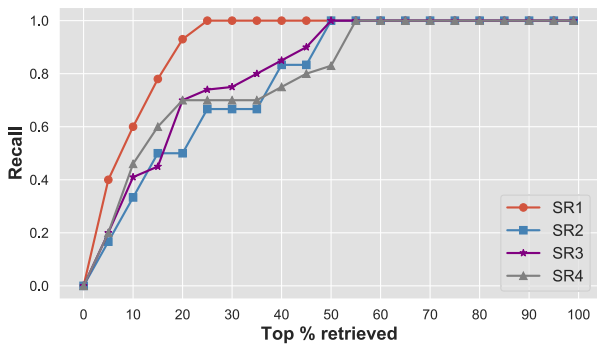


Fig. 4. Recall performance curve of 4 systematic reviews (SR). The details of four SRs can be found in Table 4.

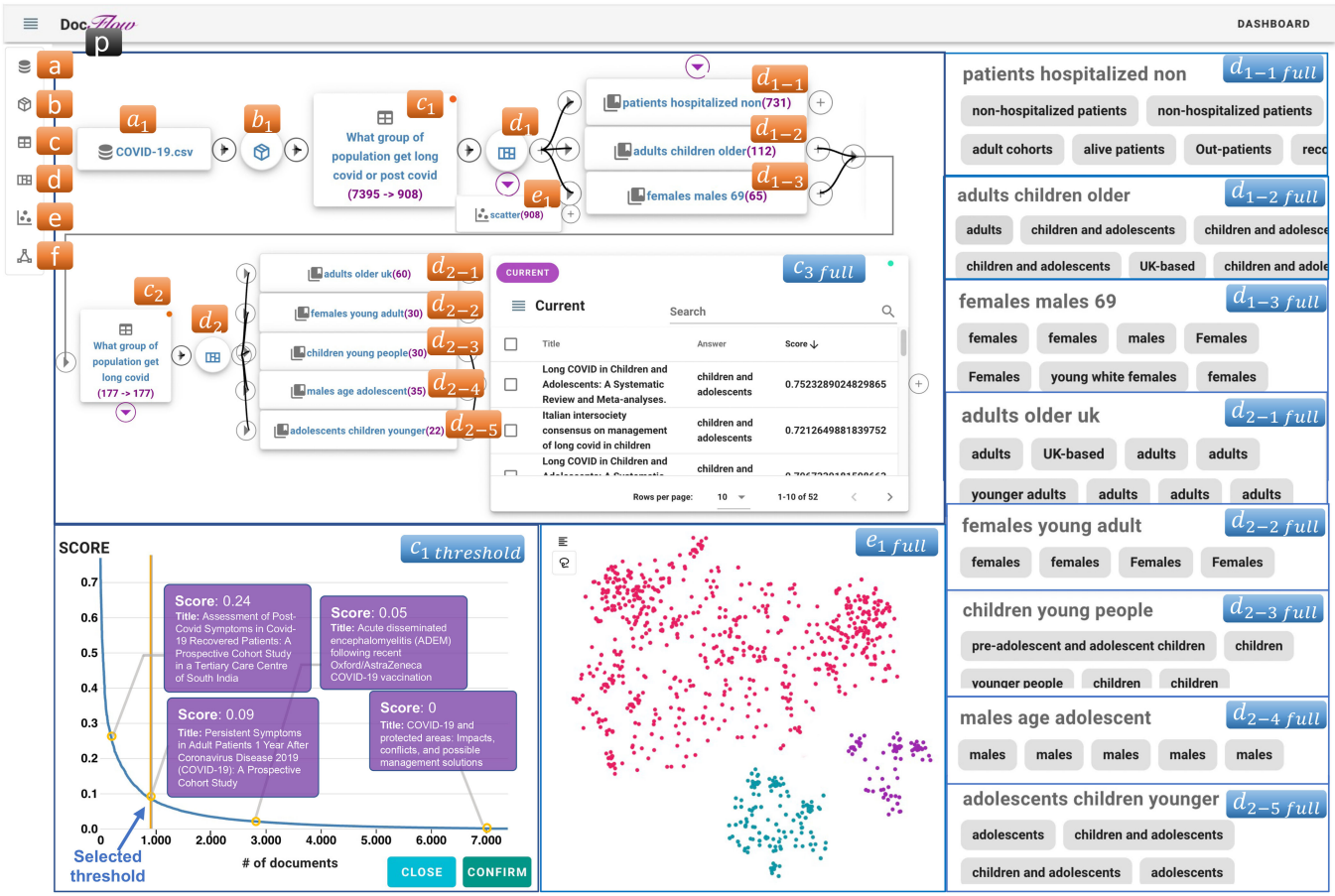


Fig. 5. Discovered insights with the query *"What group of population get long covid or post covid?"* *p* is the user-constructed retrieval pipeline. *a*, a_1 is the data loader; *b*, b_1 is the embedding generator; *c*,c_1-c_3 are the document explorers. $c_{1(Threshold)}$ is the threshold view of c_1. *d*, d_1-d_2 are the document categorizers. After the categorization, each categorizer presents categories as category card like d_{1-1}, d_{1-2}, d_{2-1}.... They can be expanded to show the detail information (displayed in $d_{1-1(full)}$, $d_{1-2(full)}$...). *e*, e_1 is the document mapper. $e_{1(full)}$ is the full view of minimized e_1.

were three categories. The first category (Fig. 5d_{1-1}) was labeled as *"patients hospitalized non"*, which contained 731 out of 908 documents; the second category (Fig. 5d_{1-2}) containing 112 documents was related to ("adults children older"); and the third category containing 65 documents (Fig. 5d_{1-2}) was labeled as "females males 69". E1 found that each category covers a group of the population. However, she pointed out that the label *"patients hospitalized"* was too general. She assessed the extracted features (Fig. 5$d_{1-1(full)}$) in the category and found that the extracted features are indeed for general patients and do not contain specific population information. The second and the third categories clearly specified the population, such as "adults" and "adolescents," in the studies (Fig. 5$d_{1-2(full)}$). Since E1 attempted to classify all documents according to adult, child, female, and male, she used a document mapper to check the distribution of the feature embedding. As indicated in Fig. 5e_1, $e_{1(full)}$, the features were grouped into three clusters. The biggest cluster corresponds to the *"patient hospitalized non"* category, which confirms the automatic categorization results. Hence, she dragged in another document explorer (Fig. 5c_2) to merge the documents in the second and the third categories. She then used the query Q1 again to categorize the merged 117 documents (Fig. 5d_2). This time, she obtained five categories: *"adults older uk "*, *"female young adults"*, *"children young people"*, *"makes age adolescent"* and *"adolescents children younger"* (Fig. 5d_{2-1}-d_{2-5}). The documents previously in the same group were split into multiple granular categories. For instance, the *"adults children older"* category was partitioned into the categories of *"children young people"* and *"adults older uk"*; and the *"females males 69"* category became the categories of *"males age adolescent"* and *"females young adult"*. The result was consistent with the CDC's finding that younger adults are more likely to have long COVID. Overall, E1 was pleased with the new granular categorization results (Fig. 5$c_{3(full)}$). *Q2: what are the symptoms of long covid?* E3 followed E1's process and explored the typical symptoms associated with long covid. E3 uploaded the same document file into a document loader (Fig. 6a_1) and connected the document loader with an embedding generator (Fig. 6b_1) to start the exploration. E3 dragged in a document explorer (Fig. 6$c_{1(full)}$) with query Q2 and retrieved 864 out of 7395 documents. Through the Threshold Fine-tuning View (Fig. 6$c_{1(Threshold)}$), she found that the threshold of the relevant score is set to 0.1. The corresponding paper is titled "Persistent Symptoms in Patients Recovering From COVID-19 in Denmark," which studies the symptoms of long COVID in Denmark. Papers with scores lower than this threshold, for instance, "The cost of Quarantine: Projecting the Financial Impact of Canceled Elective Surgery on the National Hospitals," are likely describing the impact of long COVID rather than

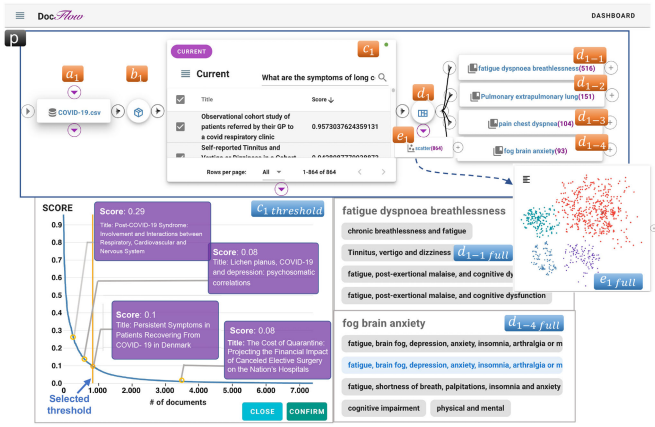


Fig. 6. The expert discovered the corpus with the query "What are the symptoms of long covid?".

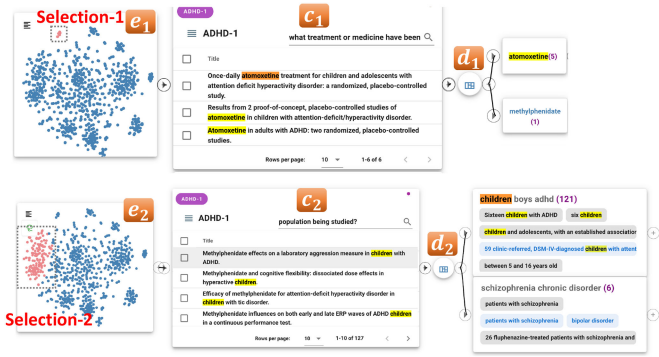


Fig. 7. Users can explore the document collection through the document mapper. e_1 and e_2 are two document mappers created to visualize document distributions. The documents, selection-1 and selection-2, are transmitted to the document explorer c_1 and c_2, and further categorized in d_1 and d_2.

symptoms. E3 then categorized the retrieved documents using the same query. Four categories showed up at the end of the document categorizer (Fig. 6d_1): *"fatigue dyspnoea breathlessness"*, *"Pulmonary extrapulmonary lung"*, *"pain chest dyspnea"* and *"fog brain anxiety"*. Based on E3's knowledge about long covid, she confirmed that the four labels were reasonable in describing symptoms. However, she wondered why the symptoms of fatigue and breathlessness were in the same category. She examined the first category and found that the identified answers were mostly a text span that included multiple symptoms, such as "chronic breathlessness and fatigue". Although an answer may contain multiple concepts, DocFlow treated the answer as a single feature. Despite of this, E3 was pleased with the extracted categories. She also confirmed the feasibility of the feature embedding and the resulting four categories. (Fig. 6e_1).

7.3.2 Systematic Review With a Customizable Pipeline

In the second case study, we demonstrate how a domain expert leverages DocFlow to quickly explore and retrieve publications (documents) and fulfill her research needs.

Data Preparation. E1 performed a systematic review of ADHD (Attention-deficit/hyperactivity disorder) drug studies. To collect the documents for exploration, E1 searched PubMed with keywords, "ADHD, medication, treatment, side-effect", and then downloaded 801 abstracts in a CSV file. These articles were in the type of "randomized controlled trials," which was considered the highest quality of the study design.

Document Exploration. E1 uploaded the documents to DocFlow using a *data loader* and then selected BioBERT in the *embedding generator* to encode both the titles and abstracts into features. By linking and observing a *document mapper* (Fig. 7e_1), she found that many documents were clustered. She then hovered over several clusters, read the document titles, and noticed that documents in the same cluster were likely to study the effect of similar drug treatments. To confirm that, she used a lasso tool in the *document mapper* to select a small cluster (Fig. 7$selection-1$) and dragged them into a *document explorer* to review the details. As shown in Fig. 7c_1, the small cluster consists of 6 studies, and most of the titles contain *"Atomoxetine"*, which is a drug approved for treating ADHD. Next, she categorized the documents in this cluster based on treatments. She queried "what treatment or drug has been used in the study" to trigger the *categorizer*. As shown in Fig. 7d_1, two categories appeared. The first category was with the label of "Atomoxetine" and contained 5 out of the 6 documents; the second category was with the label of "Methylphenidate" and contain only 1 document. After reviewing all documents, she confirmed that the categorization was correct.

E1 then applied the same process to explore another cluster (Fig. 7$selection-2$) that consisted of 127 documents. In this cluster, titles of many documents contained the words *"children"* and *"boys"* but contained different drug treatments. Accordingly, she queried "what is the population?" and connected the explorer with a document categorizer. This time, two categories appeared (Fig. 7d_2). 121 out of 127 documents were included in the first category with the label of *"children boys adhd"*. The extracted label was related to children, such as "sixteen children with ADHD" and "between 5 and 16 years old". The rest were included in the second category with the label of *"schizophrenia chronic disorder"*, which indicates patients with schizophrenia and disorder. E1 was satisfied with the results of the document mapper and categorization.

Retrieve and Categorize Documents with a Customized Pipeline. After the initial exploration, E1 decided to retrieve clinical trials studying the effectiveness of Methylphenidate on children with ADHD. She restarted a new exploration by connecting a *document mapper* with a *embedding generator* (Fig. 8c_1). Following the *PICO* [57] (*population, intervention/treatment, comparison and outcome*) framework, she queried *"what is the population?"* to trigger the question-based retrieval. E1 then added a *categorizer* to group documents into four categories: *"patients hospitalized schizophrenic"*, *"children ADHD attention"*, *"schizophrenic chronic disorder"* and *"healthy subjects volunteers"* (Fig. 8d_1), based on the query. She examined documents in the category of *"children ADHD attention"*, where the extracted answers contain *"sixteen children with ADHD," "children and adolescents"* (Fig. 8$d_{1-2full}$),.., etc. This was the population that E1 was the most interested in. Therefore, she queried and categorized documents in *"children ADHD attention"* category (143

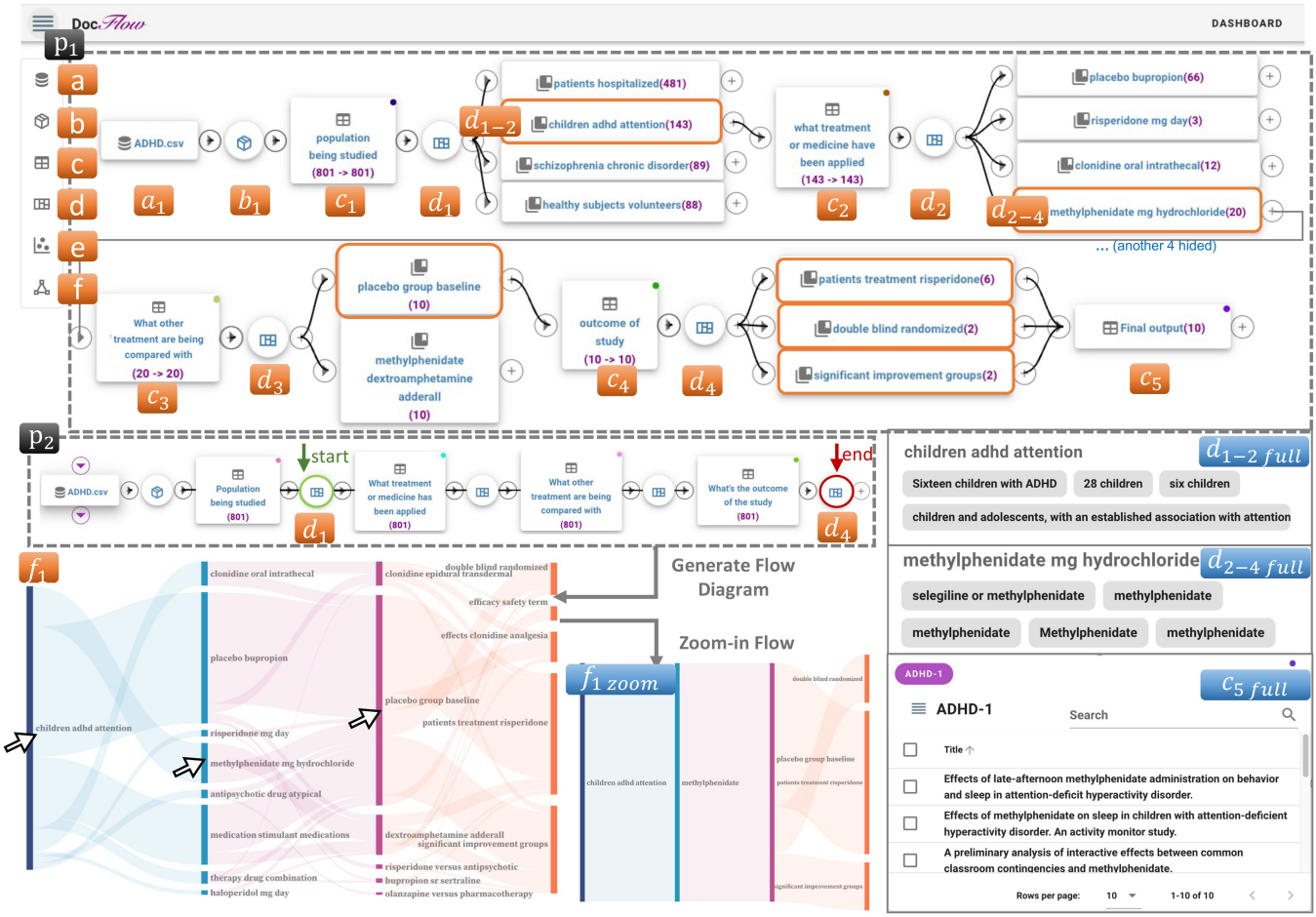


Fig. 8. A complete systematic review following the *PICO* framework on an ADHD corpus. p_1 is the retrieval pipeline constructed with 4 retrieval and categorization pairs ((c_1, d_1), (c_2, d_2),..., (c_4, d_4)), each of which corresponds to one concept in *PICO*. The final retrieved documents are transmitted to c_5 for details checking and exportation. p_2 is the sketch version of p_1 with all categories hidden; Users can specify the start and end categorizers in p_2 to zoom in on the details of the document flow as in f_1; They also can right-click the nodes in f_1 to zoom in on the flow details as in $f_{1(zoom)}$.

documents included) for further retrieval (Fig. 8d_{1-2}). As a result, 8 categories showed up, including *"placebo bupropion"* (66 documents), *"risperidone mg day"* (3 documents), *"methylphenidate mg hydrochloride"* (20 documents),..., etc. She found that the category with title *"methylphenidate mg hydrochloride"* matched her interest in treatment because it included 20 studies with *"methylphenidate"* as treatment (Fig. 8$d_{2-4(full)}$). Next, E1 added a document explorer and a categorizer after the *"methylphenidate* category. She queried *"What treatments are being compared with?"* and received 2 categories: *"placebo group baseline"* and *"methylphenidate dextroamphetamine"*. Since she wanted to review studies comparing the effectiveness of *"methylphenidate"* and *"placebo"*, she categorized the first category (10 documents included) with query *"outcomes of the study"* and received three categories. Through the process, she could identify documents related to her *PICO* questions and then studied details of interest in the retrieved documents (Fig. 8c_5, $c_{5(full)}$). E1 stated, "the categorization results are adequate" and "the system can help clinicians build a retrieval pipeline efficiently. It is very useful for systematic reviews." Meanwhile, E1 expressed her interest in categorizing documents based on multiple queries. She wanted to see the relationship between queries and categorized documents. She commented, "It will be helpful to have a visualization that can explicitly present documents in each category and how the category of documents changes given different queries."

Flow Explorer and Corpus Summarization. We implemented a flow explorer based on E1's feedback and provided her a demo that can visualize document categories with multiple queries. We first reproduced E1's retrieval pipeline (Fig. 8p_2), added a flow explorer, and then specified the start and end categorizers in the pipeline. By clicking the *"children"* label in the first dimension, paths flowing from the children category were highlighted (Fig. 8f_1). E1 found that the paths (143 documents) covered 8 different treatments in the second dimension, such as *"Clozapine"* and *"Placebo bupropion,"*. The treatments were further compared with 6 other treatments in the third dimension and resulted in 5 different outcome categories. As indicated, the flow explorer clearly summarizes the results of the retrieval pipeline in Fig. 8p_1.

With the flow explorer, E1 could quickly obtain the studies related to different treatments. By clicking "placebo bupropion" in the second dimension (Fig. 9f_1), E1 observed that the treatment had been studied on all population groups, and existing studies have already compared its performance with all other treatments. However, when E1 clicked "clozapine drug disorder" (Fig. 9f_2), she was surprised that none of the studies studying its performance on

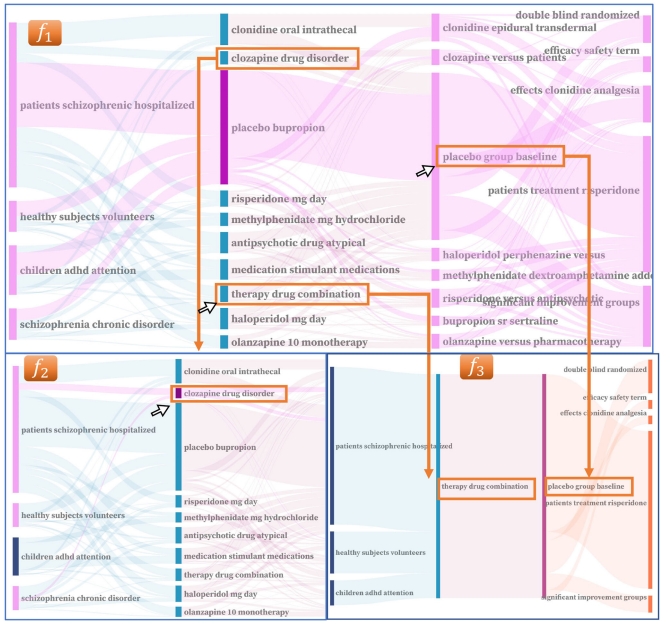


Fig. 9. The flow explorer shows the distribution of the selected documents. f_1 and f_2 show the distributions of documents that study "*placebo bupropion*" and "*clozapine*" as treatments, respectively. Users can click "*therapy drug combination*" and "*placebo group baseline*" to zoom-in the flow diagram to f_3.

children, and she was wondering if DocFlow made mistakes. After researching the literature on PubMed, she found many significant side effects of clozapine in children [58]. Hence, few clinical trials were conducted to treat children with ADHD. Because of the finding, E1 appreciated the usability of the flow explorer. She commented, "the flow explorer provides a clear summarization of document categories. It reveals the existing research directions, and making it intuitive for clinicians to understand, confirm, and be inspired to new questions and directions."

Flexible Retrieval via Flow Explorer. Clinicians could examine the categorized documents of a retrieval pipeline in the flow explorer. For example, E1 clicked the *children* node on the population dimension, *methylphenidate* node as the treatment, *placebo* node as the comparison. The zoomed-in view of the flow diagram shows the distribution of 10 articles in the subgroup (Fig. 8$f_{1(zoom)}$). After reading the retrieved studies, E1 realized the effectiveness of *methylphenidate*. She was then curious about the performance of "*drug combination* with *placebo*" on different populations and check how many studies have been done in this direction. Thanks to the flow explorer, E1 did not have to rebuild a new retrieval pipeline. Instead, she could clicked the "*therapy drug combination*" node in the second dimension, and the "*placebo group baseline*" as the compared group (Fig. 9f_1), the zoomed-in view of the diagram (Fig. 9f_3) displayed the selected subgroup of articles (27 articles) and their distribution on population and final outcome. E1 found that the comparison between *therapy drug combination* and *placebo baseline* has been studied on three out of four population groups and resulted in five different outcomes. She then exported all these documents for studying details.

Given the operations and the observed insights, E1 and E3 highly appreciated the flexibility of DocFlow. Our approach demonstrated that the information retrieval process can be flexible, efficient, and reproducible.

8 Discussion

Performance of Systematic Reviews Using DocFlow. Our question-based retrieval reduces the load on clinicians by reducing the average top-retrieval rate from 65.8% to 43% [56]. However, Fig. 4 shows that clinicians still have to review almost half of the documents if they do not want to miss any relevant document when using DocFlow. We discuss this issue with our domain expert and summarize the following reasons. First, different results might arise in a systematic review process due to different judgments by domain experts about how to identify studies, which studies to include, which data to collect, and how to aggregate results [59], [60], [61], [62]. Even domain experts have different opinions on difficult cases. Second, the number of related publications has increased. Although we followed the strategy indicated in the SR paper to collect documents, we obtained a larger document set. For the example in SR1, we obtained 1198 articles from PubMed, whereas the search in the original study yielded only 811 documents. Our retrieved documents could cover up-to-date and relevant publications not included in the original SR paper, which leads to a relatively low recall. Despite the low recall, DocFlow contains not only a retrieval function but also a question-based document categorization tool. Clinicians can categorize the retrieved documents before reading them, significantly reducing their burden.

Transferability and Extensibility. We have demonstrated DocFlow in the biomedical domain. It also can benefit other domains by using the domain-specific encoder. Specifically, by replacing BioBERT with SciBERT [63], which is pretrained on scientific publications, DocFlow can support scientists in executing a literature review. In addition, from the system perspective, extending DocFlow is simple – by adding new components that can facilitate document processing and visualization since it is a component-based system.

*Limitations and Future Work.*Although we have demonstrated the effectiveness of DocFlow in retrieving and categorizing documents, there is still ample space to improve. First, DocFlow shares a common issue with other recommendation systems: clinicians might not know what they are missing if they only focus on top-ranked documents. Although clinicians can adjust the thresholds in DocFlow to investigate the lower-ranked results, the increasing loads would prevent them from doing so. Second, DocFlow generates the label of each category based on the answers extracted from documents, which are a sequence of words and can be further separated into finer granularity. For example, DocFlow treats "chronic breathlessness and fatigue" (Fig. 6$d_{1-1full}$) as a label, yet the label contains three different symptoms. Extending DocFlow with entity detection and categorizing documents based on entity-level granularity will be helpful. Third, the computation and memory consumption of DocFlow is proportional to the number of documents. The system runs smoothly on a modern computer for tasks within 20,000 documents. The scalability can be improved if clinicians attempt to use DocFlow to process the corpus with more documents. In the future, we will work closely with domain experts and extend DocFlow to a better literature exploration system.

9 Conclusion

We have introduced a visual analytics system, DocFlow, to facilitate systematic reviews in the biomedical domain. DocFlow enables users to retrieve and categorize documents with natural language questions and question-format terms. Two BioBERT-based models are proposed to support efficient retrieval and categorization. In addition, DocFlow is a component-based system in which each retrieval, categorization, and visualization component is reusable. Clinicians can construct the pipeline of a systematic review by manipulating the components flexibly. The scatter plots and Sankey diagrams are applied to visualize the distribution and the relationships between documents, respectively, after the documents are hierarchically retried and categorized. For evaluation, we conducted two case studies and quantitative measurements. The results demonstrated the usefulness and effectiveness of DocFlow.

References

[1] E. N. M. B. McClellan, J. M. McGinnis and L. Olsen, *Evidence-Based Medicine and the Changing Nature of Health Care: Meeting Summary IOM Roundtable on Evidence-Based Medicine*, Washington, DC, USA: National Academies Press, 2008.

[2] P. Borlund, "Interactive information retrieval: An introduction," *J. Inf. Sci. Theory Pract.*, vol. 1, Sep. 2013, Art. no. 3.

[3] G. Amati, *BM25*, pp. 257–260, Berlin, Germany: Springer US, 2009.

[4] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1188–1196.

[5] A. M. Dai, C. Olah, and Q. V. Le, "Document embedding with paragraph vectors," 2015, *arXiv:1507.07998*.

[6] J. Lee et al., "BioBERT: A pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, 2019.

[7] P. Federico, F. Heimerl, S. Koch, and S. Miksch, "A survey on visual approaches for analyzing scientific literature and patents," *IEEE Trans. Vis. Comput. Graph.*, vol. 23, no. 9, pp. 2179–2198, Sep. 2017.

[8] P. Isenberg et al., "Vispubdata.org: A metadata collection about ieee visualization (VIS) publications," *IEEE Trans. Vis. Comput. Graph.*, vol. 23, no. 9, pp. 2199–2206, Sep. 2017.

[9] N. J. van Eck and L. Waltman, "Citation-based clustering of publications using citnetexplorer and vosviewer," *Scientometrics*, vol. 111, pp. 1053–1070, May 2017.

[10] F. Heimerl, Q. Han, S. Koch, and T. Ertl, "Citerivers: Visual analytics of citation patterns," *IEEE Trans. Vis. Comput. Graph.*, vol. 22, no. 1, pp. 190–199, Jan. 2016.

[11] J. Portenoy, M. Radensky, J. D. West, E. Horvitz, D. S. Weld, and T. Hope, "Bursting scientific filter bubbles: Boosting innovation via novel author discovery," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2022, pp. 1–13.

[12] P. Isenberg, T. Isenberg, M. Sedlmair, J. Chen, and T. Möller, "Visualization as seen through its research paper keywords," *IEEE Trans. Vis. Comput. Graph.*, vol. 23, no. 1, pp. 771–780, Jan. 2017.

[13] Y. Wu, X. Jin, and Y. Xue, "Evaluation of research topic evolution in psychiatry using co-word analysis," *Medicine*, vol. 96, Jun. 2017, Art. no. e7349.

[14] O. Barbosa, R. dos Santos, and D. Viana, "EvidenceSET: A tool for supporting analysis of evidence and synthesis of primary and secondary studies," 2017.

[15] E. K. Lee and K. Uppal, "Cerc: An interactive content extraction, recognition, and construction tool for clinical and biomedical text," *BMC Med. Informat. Decis. Mak.*, vol. 20, Dec. 2020, Art. no. 306.

[16] A. G. Dias, E. E. Milios, and M. C. F. de Oliveira, "TRIVIR: A visualization system to support document retrieval with high recall," in *Proc. ACM Symp. Document Eng.*, 2019, pp. 1–10.

[17] A. Narechania, A. Karduni, R. Wesslen, and E. Wall, "VITALITY: Promoting serendipitous discovery of academic literature with transformers & visual analytics," *IEEE Trans. Vis. Comput. Graph.*, vol. 28, no. 1, pp. 486–496, Jan. 2021.

[18] X. Pocco, J. Poco, M. Viana, R. de Paula, L. G. Nonato, and E. Gomez-Nieto, "DRIFT: A visual analytic tool for scientific literature exploration based on textual and image content," in *Proc. SIBGRAPI Conf. Graph., Patterns Images*, 2021, pp. 136–143.

[19] X. Ji, R. Machiraju, A. Ritter, and P.-Y. Yen, "Examining the distribution, modularity, and community structure in article networks for systematic reviews," *Proc. AMIA Annu. Symp. Proc.*, 2015, pp. 1927–1936.

[20] N. Cao, J. Sun, Y.-R. Lin, D. Gotz, S. Liu, and H. Qu, "FacetAtlas: Multifaceted visualization for rich text corpora," *IEEE Trans. Vis. Comput. Graph.*, vol. 16, no. 6, pp. 1172–1181, Nov./Dec. 2010.

[21] F. Heimerl, M. John, Q. Han, S. Koch, and T. Ertl, "Docucompass: Effective exploration of document landscapes," in *Proc. IEEE Conf. Vis. Analytics Sci. Technol.*, 2016, pp. 11–20.

[22] M. Kim, K. Kang, D. Park, J. Choo, and N. Elmqvist, "Topiclens: Efficient multi-level visual topic exploration of large-scale document collections," *IEEE Trans. Vis. Comput. Graph.*, vol. 23, no. 1, pp. 151–160, Jan. 2017.

[23] A. Rossanez, J. C. dos Reis, R. d. S. Torres, and H. de Ribaupierre, "Kgen: A knowledge graph generator from biomedical scientific literature," *BMC Med. Informat. Decis. Mak.*, vol. 20, Dec. 2020, Art. no. 314.

[24] A. Ponsard, F. Escalona, and T. Munzner, "Paperquest: A visualization tool to support literature review," in *Proc. CHI Conf. Extended Abstr. Hum. Factors Comput. Syst.*, 2016, pp. 2264–2271.

[25] A. Dattolo, M. Corbatto, and M. Angelini, "Authoring and reviewing bibliographies: Design and development of a visual analytics online platform," *IEEE Access*, vol. 10, pp. 21631–21645, 2022.

[26] K. Choe, S. Jung, S. Park, H. Hong, and J. Seo, "Papers101: Supporting the discovery process in the literature review workflow for novice researchers," in *Proc. IEEE Pacific Visual. Symp.*, 2021, pp. 176–180.

[27] M. Dehghani, H. Zamani, A. Severyn, J. Kamps, and W. B. Croft, "Neural ranking models with weak supervision," in *Proc. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2017, pp. 65–74.

[28] G. Zheng and J. Callan, "Learning to reweight terms with distributed representations," in *Proc. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2015, pp. 575–584.

[29] Z. Dai and J. Callan, "Context-aware sentence/passage term importance estimation for first stage retrieval," 2019, *arXiv: 1910.10687*.

[30] P.-S. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck, "Learning deep structured semantic models for web search using clickthrough data," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2013, pp. 2333–2338.

[31] Y. Shen, X. He, J. Gao, L. Deng, and G. Mesnil, "A latent semantic model with convolutional-pooling structure for information retrieval," in *Proc. ACM Conf. Inf. Knowl. Manage.*, 2014, pp. 101–110.

[32] B. Mitra, E. Nalisnick, N. Craswell, and R. Caruana, "A dual embedding space model for document ranking," 2016, *arXiv:1602.01137*.

[33] M. Pai et al., "Systematic reviews and meta-analyses: An illustrated, step-by-step guide," *Nat. Med. J. India*, vol. 17, no. 2, pp. 86—95, 2004.

[34] M. Martinic, D. Pieper, A. Glatt, and L. Puljak, "Definition of a systematic review used in overviews of systematic reviews, meta-epidemiological studies and textbooks," *BMC Med. Res. Methodol.*, vol. 19, 11 2019, Art. no. 203.

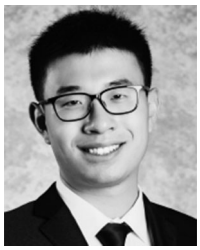
[35] L. Manchikanti, R. Derby, L. Wolfer, V. Singh, S. Datta, and J. A. Hirsch, "Evidence-based medicine, systematic reviews, and guidelines in interventional pain management: Part 7: Systematic reviews and meta-analyses of diagnostic accuracy studies," *Pain Physician*, vol. 12, no. 6, pp. 929—963, 2009.

[36] A. M. Cohen, W. R. Hersh, K. Peterson, and P.-Y. Yen, "Reducing workload in systematic review preparation using automated citation classification," *J. Amer. Med. Informat. Assoc.*, vol. 13, no. 2, pp. 206–219, 2006.

[37] J. Yang, A. Cohen, and M. McDonagh, "SYRIAC: The systematic review information automated collection system a data warehouse for facilitating automated biomedical text classification," in *Proc. AMIA Symp.*, 2008, pp. 825–829.

[38] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[39] H. A. M. Hassan, G. Sansonetti, F. Gasparetti, A. Micarelli, and J. Beel, "BERT, ELMO, USE and InferSent sentence encoders: The panacea for research-paper recommendation?," in *13th ACM Conf. Recommender Syst.*, 2019, pp. 6–10.
[40] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019.
[41] I. Beltagy, K. Lo, and A. Cohan, "SciBERT: A pretrained language model for scientific text," in *Proc. Conf. Empir. Methods Natural Lang. Process. Int. Joint Conf. Natural Lang. Process.*, 2019, pp. 3606–3611.
[42] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with GPUs," 2017, *arXiv:1702.08734*.
[43] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, *arXiv:1807.03748*.
[44] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000 questions for machine comprehension of text," 2016, *arXiv:1606.05250*.
[45] M. Joshi, E. Choi, D. S. Weld, and L. Zettlemoyer, "Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension," 2017, *arXiv:1705.03551*.
[46] T. Kwiatkowski et al., "Natural questions: A benchmark for question answering research," *Trans. Assoc. Comput. Linguistics*, vol. 7, pp. 453–466, 2019.
[47] A. Pampari, P. Raghavan, J. Liang, and J. Peng, "emrQA: A large corpus for question answering on electronic medical records," 2018, *arXiv:1809.00732*.
[48] H. Humaira and R. Rasyidah, "Determining the appropiate cluster number using elbow method for K-means algorithm," in *Proc. 2nd Workshop Multidisciplinary Appl.*, 2020.
[49] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform manifold approximation and projection for dimension reduction," 2018, *arXiv:1802.03426*.
[50] P. Riehmann, M. Hanfler, and B. Froehlich, "Interactive sankey diagrams," in *Proc. IEEE Symp. Inf. Visual.*, 2005, pp. 233–240.
[51] C. Dreisbach, T. A. Koleck, P. E. Bourne, and S. Bakken, "A systematic review of natural language processing and text mining of symptoms from electronic patient-authored text data," *Int. J. Med. Informat.*, vol. 125, pp. 37–46, 2019.
[52] L. A. Despins, "Automated deterioration detection using electronic medical record data in intensive care unit patients: A systematic review," *Pain Physician*, vol. 36, no. 7, pp. 323–330, 2018.
[53] Y. Lee et al., "Safety and usability guidelines of clinical information systems integrating clinical workflow: A systematic review," *Healthcare Informat. Res.*, vol. 24, pp. 157–169, Jul. 2018.
[54] N. C. Munshi, and H. Avet-Loiseau, "Clinical cancer research: An official journal of the american association for cancer research," vol. 17, pp. 1234–42, 2011.
[55] L. L. Wang et al., "CORD-19: The COVID-19 open research dataset," 2020, *arXiv:2004.10706*.
[56] X. Ji, A. Ritter, and P.-Y. Yen, "Using ontology-based semantic similarity to facilitate the article screening process for systematic reviews," *J. Biomed. Informat.*, vol. 69, pp. 33–42, 2017.
[57] C. Schardt, M. B. Adams, T. Owens, S. Keitz, and P. Fontelo, "Utilization of the PICO framework to improve searching PubMed for clinical questions," *BMC Med. Informat. Decis. Mak.*, vol. 7, Dec. 2007, Art. no. 16.
[58] V. Rachamallu, B. W. Elberson, E. Vutam, and M. Aligeti, "Off-label use of clozapine in children and adolescents—A literature review," *Amer. J. Therapeutics*, vol. 26, no. 3, pp. e406–e416, 2019.
[59] J. P. Wanous, S. E. Sullivan, and J. Malinak, "The role of judgment calls in meta-analysis," *J. Appl. Psychol.*, vol. 74, no. 2, 1989, Art. no. 259.
[60] B. Tendal et al., "Disagreements in meta-analyses using outcomes measured on continuous or rating scales: Observer agreement study," *BMJ*, vol. 339, 2009, Art. no. b3128.
[61] C. Palpacuer, K. Hammas, R. Duprez, B. Laviolle, J. Ioannidis, and F. Naudet, "Vibration of effects from diverse inclusion/exclusion criteria and analytical choices: 9216 different ways to perform an indirect comparison meta-analysis," *BMC Med.*, vol. 17, no. 1, pp. 1–13, 2019.
[62] N. R. Haddaway and T. Rytwinski, "Meta-analysis is not an exact science: Call for guidance on quantitative synthesis decisions," *Environ. Int.*, vol. 114, pp. 357–359, 2018.
[63] I. Beltagy, K. Lo, and A. Cohan, "SciBERT: A pretrained language model for scientific text," 2019.

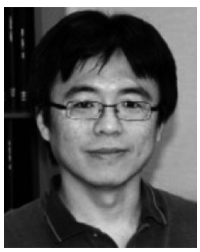


Rui Qiu received the BS degree in mathematics from the Tianjin University of Finance and Economics, and the master's degree in financial math from Washington University in St. Louis, in 2019. He is currently working toward the PhD degree in computer science and engineering with The Ohio State University. His research interests include visual text analysis, information retrieval, NLP and clinical informatics.

Yamei Tu received the BS degree in software engineering from East China Normal University. She is currently working toward the PhD degree with the Graphics & Visualization Study (GRAVITY) Research Group under Computer Science and Engineering, Ohio State University. Her research interests are visualization, text analysis, and machine learning.

Yu-Shuen Wang received the PhD degree from Visual System Laboratory, National Cheng Kung University, Tainan, Taiwan, Republic of China, in 2010. He is an associate professor with the Department of Computer Science, National Chiao-Tung University. Currently, he leads the Computer Graphics and Visualization Lab, Institute of Multimedia Engineering. His research interests include computer graphics, image manipulation, map design, data visualization, human computer interface and virtual reality. He received the Wu Da-Yu Memorial Award and NCTU EECS Outstanding Young Scholar Award, in 2016.

Po-Yin Yen received the BS degree in nursing from National Cheng Kung University, the MS degree in medical informatics from Oregon Health & Science University, and the PhD degree in nursing from Columbia University. She is an associate professor with the Institute for Informatics, Washington University School of Medicine, and Goldfarb School of Nursing, Barnes-Jewish College, BJC HealthCare. Her research focuses on human-computer interaction, workflow analysis, and data visualization to support clinical practice.



Han-Wei Shen received the BS degree from the Department of Computer Science and Information Engineering, National Taiwan University, in 1988, the MS degree in computer science from the State University of New York, Stony Brook, in 1992, and the PhD degree in computer science from the University of Utah, in 1998. He is a full professor with the Ohio State University. From 1996 to 1999, he was a research scientist with NASA Ames Research Center in Mountain View California. His primary research interests are scientific visualization and computer graphics. He is a winner of the National Science Foundations CAREER award and U.S. Department of Energys Early Career Principal Investigator Award. He also won the Outstanding Teaching award twice in the Department of Computer Science and Engineering at the Ohio State University.