PhraseMap: Attention-based Keyphrases Recommendation for Information Seeking

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

Abstract—Many Information Retrieval (IR) approaches have been proposed to extract relevant information from a large corpus. Among these methods, phrase-based retrieval methods have been proven to capture more concrete and concise information than word-based and paragraph-based methods. However, due to the complex relationship among phrases and a lack of proper visual guidance, achieving user-driven interactive information-seeking and retrieval remains challenging. In this study, we present a visual analytic approach for users to seek information from an extensive collection of documents efficiently. The main component of our approach is a PhraseMap, where nodes and edges represent the extracted keyphrases and their relationships, respectively, from a large corpus. To build the PhraseMap, we extract keyphrases from each document and link the phrases according to word attention determined using modern language models, i.e., BERT. As can be imagined, the graph is complex due to the extensive volume of information and the massive amount of relationships. Therefore, we develop a navigation algorithm to facilitate information seeking. It includes (1) a question-answering (QA) model to identify phrases related to users' queries and (2) updating relevant phrases based on users' feedback. To better present the PhraseMap, we introduce a resource-controlled self-organizing map (RC-SOM) to evenly and regularly display phrases on grid cells while expecting phrases with similar semantics to stay close in the visualization. To evaluate our approach, we conducted case studies with three domain experts in diverse literature. The results and feedback demonstrate its effectiveness, usability, and intelligence.

Index Terms—Textual Data, Machine Learning, Visual Analytics, Natural Language Processing, User-in-the-loop

1 INTRODUCTION

INFORMATION seeking of textual data is a prospering area that helps users to locate information of interest from a large corpus. Clinicians, for example, must constantly keep track of the latest research publications to perform evidence-based practice (EBP) [1]. However, even with many advanced information retrieval methods, the growing volume of information poses a significant challenge for clinicians and researchers to identify key evidence and other relevant information efficiently. For example, COVID-19 related studies have grown to over 800,000 by February 2022.

Driven by this important need, various techniques have been introduced to retrieve related information and convey insight to users from unstructured textual data [2]–[7]. Those techniques can be categorized into three groups based on the unit of focus: word level, phrase level, and paragraph level. Word level captures relatively less information due to the lack of context. For example, *matrix* is less precise compared to *matrix factorization* and *adjacency matrix*. A paragraph is a large text block that contains the most information. However, it includes many details that will not be used during retrieval, such as the details of matrix factorization. In contrast, phrases are between the word and paragraph levels, which are concise and contain precise information for document retrieval. As a result, recent research has shown that phrases can achieve better retrieval performance than words [8] and paragraphs [9].

Despite the success of existing phrase-based retrieval methods, there is still ample space for improvement. First, the relationships among phrases are not explicitly modeled, which makes it challenging to capture important contextual information from the document. Although current models can encode phrase relationships by the distance between the phrase embeddings [10]–[13], they fail to link phrases that are highly relevant but dissimilar in semantics, e.g., volume rendering & transfer function). Second, the amount of information in a corpus can be large and complex. Although phrases are considered a concise representation, involving humans in the loop for information retrieval is crucial but still challenging [14]–[16]. An intuitive visualization interface with an intelligent phrase-recommending system that can communicate with users is required to achieve efficient information seeking.

Graph is a data structure that can effectively represent information entities as nodes and entity relationships as edges. In this work, we propose PhraseMap, a graph representing information extracted from documents. The nodes of PhraseMap represent phrases, and the edges depict the relationships among the phrases. Leveraging the PhraseMap as an information representation, we demonstrate how it benefits information seeking. Specifically, the PhraseMap can provide an overview of a corpus. When researchers attempt to get familiar with the literature, they can browse the phrases shown in our PhraseMap to understand the concepts of a domain. In addition, the PhraseMap supports user-driven information-seeking. When researchers are interested in a specific phrase, more related information can be obtained by propagating through the graph. Furthermore, the desired information identified from the PhraseMap can be used to facilitate downstream tasks, e.g., document retrieval.

We utilize the attention mechanism in state-of-the-art language models (BERT) to model phrase relationships as it semantically captures context relevance. Considering the extensive number of phrases that can be extracted from literature, we introduce a navigation algorithm to help users explore the PhraseMap by identifying their phrases of interest. Specifically, given a question, our algorithm computes the relevance between the question and phrases through a question-answering (QA) model. While users

- *Y Tu, R Qiu, HW Shen are with the Department of Computer Science and Engineering, The Ohio State University, Columbus, OH, 43202. E-mail: tu.253, qiu.580, shen.94@osu.edu*
- *YS Wang is with National Chiao Tung University. E-mail: yushuen@cs.nctu.edu.tw*
- *PY Yen is with Washington University School of Medicine. E-mail: yenp@wustl.edu*

with various backgrounds may be interested in different topics, they can *accept* or *reject* the retrieved phrases. Our algorithm then recommends other relevant phrases according to their feedback by information propagation on the graph. Note that the recommended phrases are determined based on the context of sentences. They are related but not necessarily semantically similar, so users will not receive redundant information. In addition to the navigation algorithm, we present a resource-controlled self-organizing map (RC-SOM) to visualize the PhraseMap, which displays phrases evenly and regularly on a grid layout and keeps semantically similar phrases possibly appearing at nearby positions. The navigation and the PhraseMap visualization algorithms are integrated into a visual analytic system to help users seek information of interest. To demonstrate its effectiveness, we performed thorough case studies on three datasets and conducted expert interviews. We summarize our contributions as follows:

- We build an *attention-based PhraseMap* to summarize concepts in a corpus, where nodes and edges on the graph represent phrases and their relationships, respectively.
- We introduce a *navigation algorithm* to help users identify phrases on the PhraseMap, which are relevant to their questions and interests. This semantic navigation algorithm achieves intelligent user query and knowledge refinement.
- We present an *RC-SOM* to project PhraseMap on a grid layout. It avoids visual clutter and responds accordingly to users' feedback.
- We design a *visual analytic system* to illustrate our approach and enable human-in-the-loop information seeking from extensive collections of documents.
- We demonstrate the usefulness and effectiveness of our method in multiple forms of evaluation, including quantitative evaluations, case studies, comparison studies, and domain discussions.

2 Related Work

In this section, we provide an overview of the techniques that are most related to our work, including (1) attention analysis & applications, (2) text visualization, and (3) information seeking.

2.1 Attention Analysis & Applications

Attention-based models have demonstrated their effectiveness in many fields, including computer vision [17], recommendation systems [18], and natural language processing [19]. The idea was first presented by Ashish et al. [20], which greatly improved recurrent neural networks [21] on sequential data for parallel processing and enabled the training on huge datasets. As a result, several famous language models, such as BERT [13] and GPT [22] were pre-trained and released to benefit downstream tasks. Specifically, BERT contains a stack of encoders composed of a self-attention layer and a linear layer. The success of BERT comes from its self-attention layer, which considers contextual information when generating the embedding of each word. Given a sentence with n tokens, BERT generates a matrix of attention weights (refer to attention maps in the following sections) $\mathbf{A} \in \mathbb{R}^{n \times n}$ to describe the relationships among words. Such a weight indicates the importance of a contextual word to understand the meaning of a target word. Considering the effectiveness, several studies [23], [24] attempt to interpret attention networks by examining the relationships between attention weights and specific linguistic phenomena. For example, Reif et al. [25] provided evidence that attention maps contain grammatical representations by investigating how BERT represents syntax.

Attention weights model the dynamic relationships between each token and its context, which benefit many downstream tasks. For example, the attention weights between target words and their context can be used as a feature vector to perform sentiment analysis [26]–[29]. Besides, many works construct the word/phrase network based on the attention weights and apply the network to perform relation extraction [30][31], keyword extraction [32][33], and document retrieval [34]. The methods mentioned above verified the usefulness of attention to model word relationships. In our work, we exploit the attention mechanism to facilitate interactive phrase recommendations from a large corpus.

2.2 Text Visualization

Presenting large, unstructured textual information has been widely studied in visualization [35]. Among the designs, scatterplots are widely used in presenting the overall view and global patterns [36]–[39] of a corpus. They are often combined with dimensionality reduction techniques, such as PCA[40], SOM [41], [42], t-SNE [43], and UAMP [44], for projecting high-dimensional textual data into 2D space. Graph visualizations, particularly node-link diagrams, are exploited to display the relationships of textual entities explicitly and have demonstrated satisfactory performance [45]–[47]. The Voronoi diagram is also a popular visualization that encodes the text relationships into spatial information [48]–[52]. However, with a large volume of information, general users are difficult to identify underlying data patterns using these visual designs [53]–[55].

Map-based visualization has been proven to be visually appealing and user-friendly because humans are familiar with geo-maps. It utilizes distances and directions to present extensive relational information. One of the earliest works [56] extended maps as a good metaphor for presenting relational data. Later, GMap [54] was presented to convert the collaboration graph into a map that can reveal the underlying communities. This map metaphor was then widely used in many fields. In the social network, D-Map [55] exploits a map metaphor to represent user behaviors and analyze information patterns in social media. By incorporating temporal analysis, the authors further extended their work to D-Map+ [57] and R-Map [58]. In text analysis, many pioneers transform text-based graphs into maps [14]–[16]. Among the works, Fried et al. [14] introduced the map of computer science, which generates the visualization summarization of each topic in the DBLP database. It suffers the scalability problem when the volume of information increases because of the word layout. Although many approaches have been presented to visualize text data, they cannot be used in our system because we aim to sufficiently utilize the area for displaying a large corpus while preventing words from occluding each other. In addition, the text visualization should be dynamic to support efficient information-seeking.

2.3 Information Seeking

Information seeking from extensive document collection has been widely studied in text mining and visualization [59]. One way to achieve the goal is to organize documents into clusters according to the hidden "topics", also referred to as *topic modeling*, such as PLSA [2], LDA [3] and NMF [4]. Another way is to retrieve relevant information from organized documents based on queries [60], also known as *machine reading comprehension*. Our work is related to extractive question answering under this category.

Both topic modeling and machine reading comprehension methods transform the raw text into semantic representations for similarity comparison. Traditionally, they construct representations using lexical information (i.e., bag of words) [5]), which indicates the frequency of each keyword occurring in a document. Since keywords are too coarse to explain topics in many domain-specific applications, follow-up works improve the quality of topic modeling by incorporating phrases or n-grams into topic explanations [6][7]. Recent methods transform documents into semantic representations using advanced language models, such as BERT [13], ALBERT[11], and RoBERTA [12]. With the development of pre-trained language models, the methods leveraged its superior semantic understanding to perform extractive QA tasks [61]. However, modeling the semantic similarity as the only relationship among words will limit the extensiveness of information-seeking because relevant phrases are not necessarily similar in semantics. It motivates us to apply attention to model phrase relationships explicitly from contextual text and prevent recommending redundant phrases to users.

3 Requirement Analysis

Our work is motivated by the growing need to retrieve information of interest from a large corpus to support evidence-based practice. To better understand the domain requirements, we have collaborated closely with three domain experts, E1, E2, and E3. E1 has a clinical background with extensive systematic review experiences in the biomedical domain. E2 has a computer science background, focusing on computer graphics and visual analytics. E3 also has a background in computer science, specializing in machine learning, human-computer interface, and data visualization. The following requirements are summarized from (1) weekly discussions with domain experts and (2) a literature review of the existing visual analytic approach for information seeking:

R1. Information extraction and transformation from a large corpus. It takes time for users to process a large corpus and organize useful information. All domain experts agree that automatic data transformation from documents to a structured format eases the burden on users and benefits many downstream tasks. The extracted data representation should:

- *R1.1 Cover all useful information.* Because users will search for evidence and perform evidence-based practices, all important data should remain in the representation [62], [63].
- *R1.2 Contain precise information. E1* and *E3* mentioned that keywords alone could not convey a concept accurately due to the lack of context. For example, *neural* could describe stem cells in a brain as well as networks that have been widely used in machine learning. The word itself is vague without context.
- *R1.3 Capture contextual phrase relationships.* All experts pointed out that contextual phrases are the key to information seeking and exploration because they co-exist to describe a concept. For example, "topic modeling" is a method of "text analysis." They should be well captured in the extracted data representation.

R2: Providing custom recommendations and navigation. Given the massive amount of extracted information, it is essential to provide users with proper guidance and keep them in the loop during information seeking. Otherwise, users could be overwhelmed by the overloaded information. It should allow the following:

- *R2.1 Recommending phrases based on users' queries.* Understanding users' needs and automatically locating the relevant information is essential. E1 indicated that exploration would be much more efficient if we enable semantic filtering to highlight the relevant information.

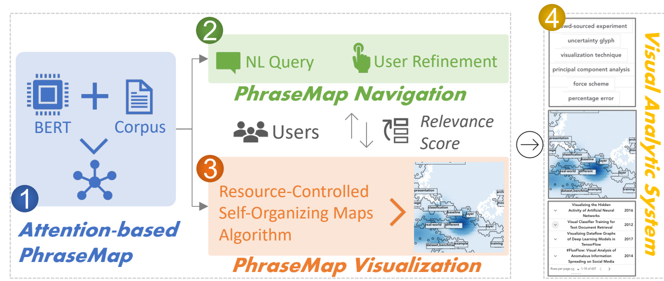


Fig. 1: Our approach consists of three steps to enable interactive information seeking of a large corpus: (1) attention-based PhraseMap construction, (2) enabling user NL query and refinements for PhraseMap navigation, and (3) grid-based layout of PhraseMap with recommendations. (4) an interactive system is developed to demonstrate our approach.

- *R2.2 Updating phrases based on users' feedback.* Users play critical roles in the visual analysis of text data. It is necessary to incorporate users' feedback to update the recommendation and steer the navigation process [36], [64], [65].

R3: Effective communication between humans and machines. The domain experts expressed the need to explore a large set of documents through an intuitive visualization interface. It can be decomposed into two specific tasks.

- *R3.1 Presenting information to users clearly.* Visualizing the data representation can present an overview of the corpus to help users understand essential concepts. However, avoiding information overload is a key issue due to the large volume of information, as pointed out by E3.
- *R3.2 Responding to users' interactions accordingly.* E2 mentioned that once users provide feedback on recommended information, the visualization should reflect the updated results immediately.

4 Approach Overview

The aforementioned requirements motivate us to develop a visual analytic approach to achieve interactive information seeking from a large corpus. The overview of our approach is shown in Figure 1. It consists of four components:

- **Attention-based PhraseMap:** We propose an attention-based phrase graph, called PhraseMap, (R1.1) to represent information for a large corpus. The basic elements of our PhraseMap are phrases because they are more informative than words and can capture contextual information from documents (R1.2). As for the edges, we utilize an advanced language model, BERT, to extract the attention scores and model the phrase relationship.
- **Navigation of PhraseMap:** To help users explore the PhraseMap, we propose a navigation algorithm to recommend relevant phrases given users' interests. Specifically, we compute the relevance score of each phrase to users' natural language queries through a question-answering model (R2.1). Later, the scores can be further updated by propagating attention scores on the PhraseMap, triggered by the user's operations (R2.2).
- **Grid-based Visualization of PhraseMap:** Visualization is a bridge to connect our PhraseMap and users. To present an extensive amount of information, we develop a resource-controlled self-organizing map algorithm to project the PhraseMap onto a grid layout. It resolves the visual clutter issue by increasing

space utilization (R3.1). Also, it can help users easily identify those relevant phrases by highlighting the grids (R3.2).

- **A Visual Analytic System:** To demonstrate the usefulness of our visual analysis approach, we design and develop a visual interactive system. The core component is the visualization of PhraseMap. Two other supportive views further help users to do visual analysis over the PhraseMap.

5 METHOD

In this section, we introduce our visual analytic approach to illustrate (1) how to construct the PhraseMap to enable interactive information seeking, (2) how to query the PhraseMap given users' interests, (3) how to visualize the PhraseMap with query results clearly and intuitively.

5.1 Attention-based PhraseMap Construction

The main challenge of constructing an effective representation for a large corpus is how to model meaningful relationships among phrases for information seeking. Semantic similarity is a common way to describe the phrase relationships. However, semantically similar phrases may have too much redundancy, which is not helpful during information seeking. For example, recommending "ray tracer" does not add much information given "ray tracing." It is more important to recommend semantically related but dissimilar phrases to reduce information redundancy, e.g., given "ray tracing," "volume rendering" expands the scope of the search so that it can be a better choice than "ray tracer." To serve this need, we propose to utilize the attention mechanism in the Transformer-based language model. The goal of the attention is that each word assigns more attention to its important contextual words. By leveraging attention as relationships, the PhraseMap can capture semantic-related information.

For a sentence with n tokens, the attention map is a $n \times n$ matrix to describe how much attention each token should allocate to all the others. To construct the PhraseMap, we need to transform the token relationships into phrase relationships. In summary, given a large corpus, the construction of the PhraseMap consists of four steps: (1) fine-tuning the BERT when there is a domain shift between pre-training datasets and application datasets. (2) extract word relationships based on token relationships from BERT, (3) transform the word relationships into phrase relationships, and (4) aggregate phrase relationships of all documents into the PhraseMap. We describe the details in the following paragraphs and illustrate the steps in algorithm 1.

Fine-tuning the BERT. Although there have been several pre-trained Transformer-based models released for public use. It should be fine-tuned on domain-relevant documents to capture the relationships of domain-specific words better, as proved in [32]. In this study, we utilize three BERT variants to construct the PhraseMap in three different literature: *Visualization, Machine Learning (ML), and Biomedical*. To learn domain knowledge in visualization, we fine-tune the BERT model on the *VisPub* dataset by classifying each abstract into its corresponding category. We directly utilize the SciBERT model [66], and BioBERT [67] for the other two literature as the pre-trained datasets of the two models are in the same literature with our focus. It is easy to adapt to other language models as long as the attention mechanism is utilized.

Word Relationships. We use the BERT-base model, which is composed of 12 layers. Each layer contains 12 attention heads in parallel to capture different features. As indicated in recent studies [68]–[71], the middle and the latter attention layers capture the syntactic information and high-level semantics, respectively. For each document d_k, we feed it into the BERT model. Then we average over the attention heads in the last layer to obtain the token relationships. Note that the BERT tokenizer splits words into subwords to limit the size of vocabulary, such as "surfboard" to "surf" and "*##board*". We automatically detect these cases and average the corresponding rows/columns in the attention map to generate the word relationships, denoted as $\mathbf{A}$, where $\mathbf{A} \in \mathbb{R}^{n \times n}$ and n is the number of words.

Algorithm 1: Constructing PhraseMap for corpus

Input: a set of documents D, with each document $d_k \in D$, $d_k = (w_1, w_2, .., w_n)$ has n tokens w_i
Output: PhraseMap **P**=(**V**, **E**), with nodes **V** and edges **E**
1 Initialize the PhraseMap **P**
2 **for** $d_k \in D$ **do**
3   averaged word relationship matrix $\mathbf{A} \leftarrow$ BERT(d_k)
4   phrase set with m phrases $(q_1,...,q_m) \leftarrow POS\_Tag(d_k)$
5     $\mathbf{T} \in \mathbb{R}^{m \times n}$, $t_{i,j} = 1$ if q_i contains w_j, otherwise $t_{i,j} = 0$
6   phrase relationship matrix $\mathbf{M} \in \mathbb{R}^{m \times n} \leftarrow \mathbf{T} \cdot \mathbf{A} \cdot \mathbf{T}^{\top}$
7   **for** m_{ij} *in* $\boldsymbol{M}$ **do**
8   **if** *edge* $(q_i \rightarrow q_j) \in \mathbf{E}$ **then**
9             $(q_i \rightarrow q_j).attn\_sum$ += m_{ij}
10             $(q_i \rightarrow q_j).fre$ += 1
        else
11   add $(q_i \rightarrow q_j)$ to **E**
12             $(q_i \rightarrow q_j).attn\_sum = m_{ij}$
13             $(q_i \rightarrow q_j).fre = 1$
        end
    end
end
14 **for** $e \in E$ **do**
15     $e.attn\_avg \leftarrow$ e.$attn\_sum$ / $e.fre$
end

Phrase relationships. To aggregate the word relationships **A** to phrase relationships **M**, we need to first extract phrases from the documents. As shown in algorithm 1, for each document, we generate the Part-Of-Speech (POS) tagging of each word. Then, we extract phrases based on a set of rules (*line 4*) because the ruled-based method is superior in its high interpretability and direct transferability from domain experts' knowledge [72], [73]. Besides, the rule-based method does not require datasets to optimize hyperparameters in the algorithm [74]–[76]. In Table 1, we summarize our rules to identify phrases and provide corresponding examples for illustration purposes. After extracting m phrases

Rule	#Token	Example
NN*-NN*	2	*volume rendering*
NN*-NN*-NN*	3	*topic modeling algorithm*
NN*-HYPH-NN*-NN*	4	*node-link diagram*
NN*-HYPH-VBN-NN*	4	*matrix-based visualization*
JJ-NN*-NN*	3	*nonnegative matrix factorization*
JJ-HYPH-NN*-NN*	4	*real-world dataset*

TABLE 1: POS-tag-based rules to extract phrases and examples. $NN*$ includes NN, NNS, and NNP.

from one document, we construct a matrix $\mathbf{T} \in \mathbb{R}^{m \times n}$ to show how

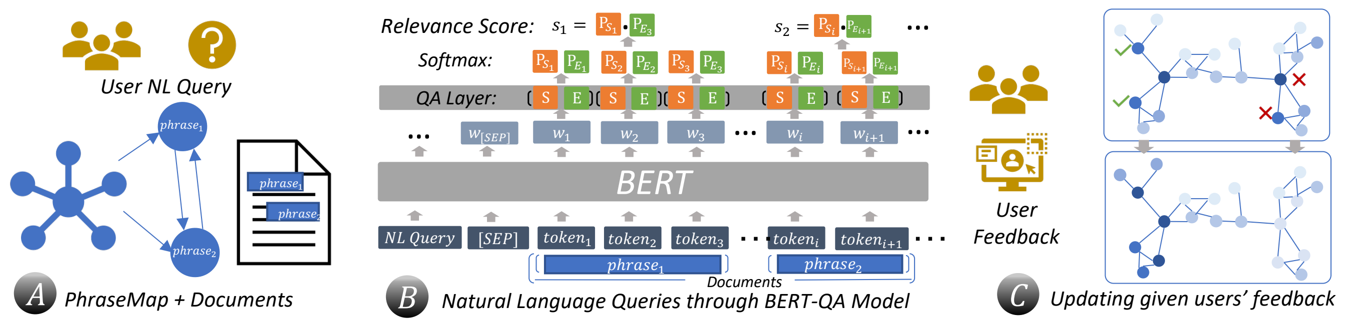


Fig. 2: The overview of PhraseMap Navigation: (A) The algorithm takes users' natural language query, the PhraseMap, and their original documents as input. (B) a BERT-QA model is employed to compute the relevance of phrases to users' queries. (C) To allow user-in-the-loop information seeking, we take users' feedback, whether accepting/declining recommended phrases, to update the relevance of phrases. It is achieved by information propagation on the PhraseMap.

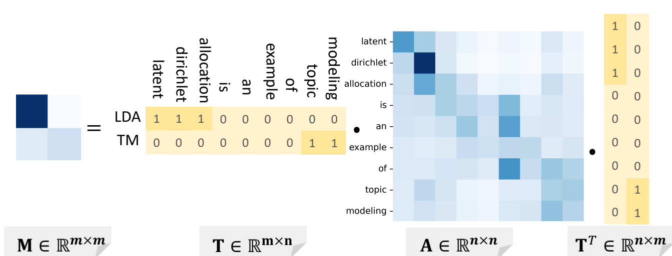


Fig. 3: Computation process of phrase PhraseMap$\mathbf{M}$ by matrix multiplication of matrix $\mathbf{T}$, word PhraseMap$\mathbf{A}$ and matrix $\mathbf{T}^\top$.

m phrases are composed of n words. For each $t_{i,j} \in T$, we use $t_{i,j} = 1$ to indicate that phrase i contains the word j and $t_{i,j} = 0$ otherwise (*line 5*). Accordingly, the phrase relationships $\mathbf{M}$ can be computed using $\mathbf{M} = \mathbf{T} \cdot \mathbf{A} \cdot \mathbf{T}^\top$(*line 6*). We point out that we retain incoming and outgoing attention when extending the relationships from words to phrases. Namely, $\mathbf{T} {\cdot} \mathbf{A} \in \mathbb{R}^{m \times n}$ represents how m phrases assign attention to n words, and by multiplying $\mathbf{T}^\top$, the resulting matrix $\mathbf{M}$ shows how the attention flows between phrases. Figure 3 shows an illustration. The input sentence is *"Latent Dirichlet Allocation is an example of topic modeling."* Two identified phrases are *"Latent Dirichlet Allocation (LDA)"* and *"topic modeling (TM)"*. Matrix $\mathbf{A} \in \mathbb{R}^{7\times 7}$ shows how each word assigns attention to other words. Matrix $\mathbf{T} \in \mathbb{R}^{2\times 7}$ indicates how the two phrases are composed using the seven words. The final matrix $\mathbf{M} \in \mathbb{R}^{2\times 2}$ is an asymmetric matrix because incoming and outgoing attention is different, indicating *LDA* does not allocate the same attention to *TM* as *TM* does to *LDA*.

PhraseMap. Taking matrix $\mathbf{M} \in \mathbb{R}^{m\times m}$ as the adjacency matrix of a directed graph, the graph has m nodes, and each $cell(i, j)$ indicates the attribute of edge from node i to node j. We merge the graphs of all documents in a corpus to create the PhraseMap. The merging process is iterative. Specifically, when merging two graphs, we first check whether an edge exists. If it does, we update the edge attributes (*line 9-10*) by increasing the attention scores and the counter. Otherwise, we add this new edge into the PhraseMap and initialize the edge attributes (*line 11-13*). After all the graphs are merged, we compute the average attention of each edge as attribute *attn_avg* (*line 15*).

5.2 PhraseMap Navigation

We help users navigate our PhraseMap to explore the information of interest. The assistance is essential since phrases extracted from a corpus can be many such that users would have no idea where to start the exploration. As visualized in Figure 2, initially, we let users ask a question and then recommend phrases that could answer the question. Because of various backgrounds, users may be interested in a part of different recommended phrases. Our navigation algorithm updates the recommended phrases based on users' feedback. We describe details of this user-in-the-loop information seeking as follows.

Phrase Recommendation based on Questions. Users start the information seeking on our PhraseMap by providing a query. An intuitive approach to realize this requirement is a word matching of all phrases on the PhraseMap, such as returning "topic modeling" if the word "topic" is queried. However, this approach often returns inaccurate phrases unless users can provide precise keywords in the query. An example is "what are the methods proposed for image classification?" The word matching method will never return the answer "convolution neural networks".

We utilize the BERT question-answering (QA) model to process natural language (*NL*) queries. Given a document and a *NL* question, the BERT-QA model computes the probability of each span in the document that can answer the question. We refer to the probability as a *relevance score*. In other words, when an *NL* query is given, we process the documents in the literature (i.e., which is used to construct the PhraseMap) and determine the relevance of each phrase on the PhraseMap. We point out that the strategy has two advantages: (1) NL allows users to express their idea accurately. (2) The BERT-QA is the latest language model that considers contextual words in sentences and can accurately determine the relevance of a phrase to the question.

As visualized in Figure 2B, the BERT-QA model takes a document and an *NL* question as input and determines the probability of each word/token that can answer the question. Specifically, the BERT-QA model is composed of a BERT encoder and a QA layer. The encoder first encodes each token i into a d dimensional embedding w_i. Then, the QA layer, which contains two sets of learnable weights $\mathbb{S}$ and $\mathbb{E}$, computes the dot products of each embedding w_i and the weights, respectively. The dot products are then undergone a softmax layer to generate the probabilities of each token being the start and end of the answer. Let P_{S_i} and P_{E_i} be the probability of token i being the start and end of the answer.

The computation of the QA layer can be formulated as:

$$P_{S_i} = \frac{e^{\mathbb{S}\cdot\mathbf{w_i}}}{\sum_j e^{\mathbb{S}\cdot\mathbf{w_j}}}, \quad P_{E_i} = \frac{e^{\mathbb{E}\cdot\mathbf{w_i}}}{\sum_j e^{\mathbb{E}\cdot\mathbf{w_j}}}. \tag{1}$$

For each phrase in the PhraseMap, which is a span from token i to token j in its original document, the *relevance score* is defined as its probability of answering the user's question and computed as:

$$\mathcal{S} = P_{S_i} \cdot P_{E_j} \tag{2}$$

Clearly, given an *NL* query, phrases with high relevance scores to the query will be recommended to start the navigation.

We trained the BERT-QA model on public QA datasets, which contain documents and the corresponding question-answer pairs. The trained model can be deployed to process the literature in similar domains. Due to the space limit, we refer readers to our supplemental material for training details.

Updating Phrases of Interest. Users with different backgrounds would be interested in different recommended phrases. We incorporate their feedback to steer the navigation. Specifically, when users *accept* that phrase v_i is important, we also increase the relevance of its neighboring phrases (to the query). Recall that phrase relationships $\mathbf{M}$ (Section 5.1) define the edges on the PhraseMap. $\mathbf{M}_{i,*}$ indicates how phrase v_i allocates its attention to all the other phrases, which is utilized to measure how phrase v_i influences others. To implement the idea, we compute the probability of walking from v_i to other phrases by normalizing $\mathbf{M}_{i,*}$. Namely,

$$\mathbf{M}_{i,j} = \frac{exp(c \cdot \mathbf{M}_{i,j})}{\sum_{p=1}^{m} exp(c \cdot \mathbf{M}_{i,p})} \tag{3}$$

One node v_i can influence its k-hop neighbors, not only the directly connected ones. The degree of influence decreases as the distance to node v_i increases. When user *accepts/declines* a node v_i, its influence on all phrases is computed as the i-th row of $\mathbf{M}^k$:

$$I(v_i) = (\mathbf{M}^k)_{i,*} \quad \in \mathbb{R}^m \tag{4}$$

k decides the graph distance to the neighbors that will be influenced, and we set $k = 3$ in our experiment as we want to avoid weak relationships. The influence is used to update the *relevance scores* of phrases to users' interests, defined as:

$$\mathcal{S}(v_j) = \begin{cases} \mathcal{S}(v_j) + I(v_j), \ if \ feedback = \text{"}accept\text{"} \\ \mathcal{S}(v_j) - I(v_j), \ if \ feedback = \text{"}decline\text{"} \end{cases} \tag{5}$$

If users accept that one phrase is important, we increase the relevance score of the neighboring phrases. Otherwise, we decrease the relevance of the phrases.

5.3 Grid-based Visualization

We present a grid-based layout to visualize the PhraseMap and enable users to discover information of interest in literature. The visualization provides an overview of the key phrases while preventing them from occluding each other (*R3.1*). It also cooperates with our navigation algorithm and interactively highlights the phrases based on users' input to facilitate information seeking (*R3.2*).

5.3.1 Design Choice

Many visual designs have been presented to visualize graph data. The most common one is a node-link diagram, which is effective for showing the relationships among nodes. However, the information is often overloaded when a large number of nodes and links are presented. To enhance readability, another common choice is a scatterplot that encodes relationships implicitly by distances [77], [78]. Namely, it displays relevant nodes at nearby positions to avoid drawing edges. In spite of the simplification, nodes that are too close to each other also induce visual clutter due to the overlap of the displayed letters. We demonstrate this visual inefficiency problem in our case study subsection 6.5. To prevent the visual clutter mentioned above while revealing phrase relationships, we propose to map the phrases into a grid layout and arrange semantically similar phrases to appear in neighboring cells.

5.3.2 RC-SOM for Grid-based layout

We map the PhraseMap onto a 2D hexagonal grid layout, where phrases located at neighboring cells are similar in semantics. We chose hexagonal girds over squared and triangular grids because of the regularity of the grid pattern. They suffer less from the edge effect due to the low perimeter-to-area ratio and have stronger visual-appealingness. To meet the requirement *R3.1*, we develop a mapping function between the phrases and hexagonal cells, which should fulfill two principles:

- Distance between 2D cells should represent the semantic similarity of phrases defined in the high-dimensional space.
- 2D cells and phrases should have a one-to-one mapping to ensure the readability of letters when the layout is zoomed at a reasonable level.

We extend the self-organizing map (SOM) [41], [42] to a resource-controlled SOM (RC-SOM) to achieve the aforementioned principles. The SOM is an unsupervised machine learning technique that can project high-dimensional data to a low-dimensional *discrete* map while preserving the data structure. Let x_1, x_2,..., x_n, $x_i \in \mathbb{R}^{768}$ be the phrase embeddings extracted from BERT and w_1, w_2,..., $w_m \in \mathbb{R}^{768}$ be the unknown weights of grid cells, and n and m are the numbers of phrases and cells, respectively. The goal of SOM is to map each data item x_i to cell c with weight w_c such that the Euclidean distance $|x_i - w_c|$ can be minimized, as illustrated in Figure 4A. The training of SOM is composed of two steps: (1) finding a cell c for each x_i that has a minimal distance $|x_i - w_c|$, and (2) updating the weights of c and its neighbors to further minimize the distances to $|x_i$. The two steps repeat until the weights converge.

SOM may project multiple phrases to a single cell because it finds the best cell c for each phrase embedding x_i. In other words, multiple phrases will be displayed at the same position, and visual clutter occurs. To prevent this problem, we modify the SOM by (1) adding a resource attribute to each cell and (2) inverting the competition. Specifically, we define a resource attribute r_i to indicate whether a cell i is occupied (Figure 4C). In addition, our RC-SOM finds the best phrase for each cell, so that each cell will display only one phrase. We describe the details of our RC-SOM as follows.

- Step 1 *Initialization*: Given n phrase embeddings $x_1, ...x_n$, $x_i \in \mathbb{R}^{768}$, initialize m cells ($m > n$) with random weights $w_1, \ldots, w_m$, $w_j \in \mathbb{R}^k$ and set the resource attributes $r_j = 1$.
- Step 2 *Similarity Matching*: For each target cell c with $r_c > 0$, assign the unassigned data item x_i to cell c, in which the distance $|w_c - x_i|$ is minimum.

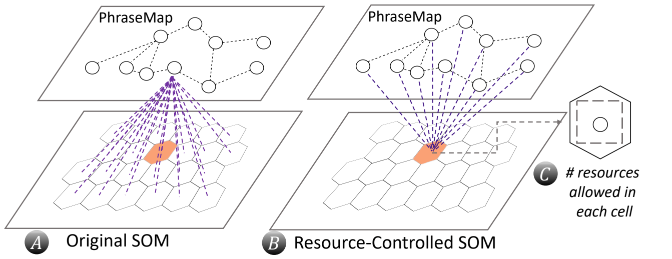


Fig. 4: Illustrations of original SOM algorithm and our RC-SOM.

- Step 3 *Weight Updating*: To achieve smooth projection results, neighboring cells are expected to have similar weights w. Hence, when cell c is matched to a specific phrase, the weight of its neighboring cell j would also be updated according to the topological distance using

$$w_j = w_j + lr(t) \cdot T_{j,c}(t) \cdot (x_i - w_j), \tag{6}$$

  where $lr(t)$ is the learning rate at epoch t, T indicates the neighborhood relationship between cells, and is defined as:

$$T_{j,c}(t) = exp\left(-\frac{d_{j,c}^2}{2\sigma(t)^2}\right). \tag{7}$$

  $d_{j,c}$ is the block distance between cells j and c, and $\sigma(t)$ controls the neighborhood size at epoch t. In our implementation, we apply the decay function

$$x(t) = f(x(0), t) = \frac{x(0)}{1 + t\tau}, \ where\ \tau = \frac{2}{E} \tag{8}$$

  to reduce the learning rate $lr(t)$ and the neighboring size $\sigma(t)$ during the learning process.
- Step 4 *Resource Updating*: Update the resource r_j to $r_j - 1$ and mark phrase x_i as assigned.
- Step 5 *Continuation*: Repeat steps 2-3 until all phrases are assigned. To start the next epoch, release all resource constraints by setting $r_j = r$, and then repeat steps 2-3.

The hyperparameters $lr(0)$ and $\sigma(0)$ could affect the performance of RC-SOM. A commonly used quality measure is quantization error [79], [80], which evaluates whether each phrase i is properly assigned to its best-matched cell c. The error is formulated as:

$$QE = \frac{1}{n}\sum_{i=1}^{n} \|x_i - w_c\|. \tag{9}$$

In our implementation, we perform a grid search with respect to lr and σ. The results with the lowest QE will be used to display the PhraseMap. Please refer to our supplemental material for the details of the grid search.

5.4 Visual Analytic System

To demonstrate our approach to achieve interactive information seeking, we develop an interactive visual analytics system. Taking the grid-based layout of the PhraseMap as our focus, several coordinated views and supportive interactions are provided to help users perform information retrieval tasks.

5.4.1 Design Tasks

We design the system based on (1) feedback from regular meetings with our domain experts to meet the requirements in section 3 (*R2-R3*) and (2) suggestions collected from literature reviews. The

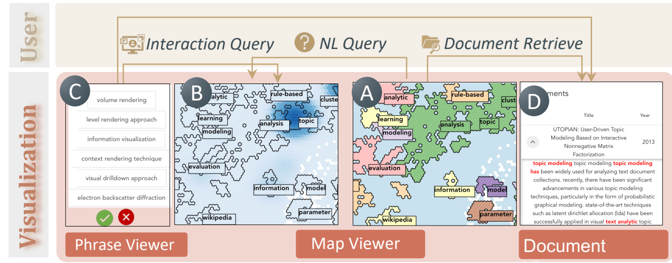


Fig. 5: The view components and workflow of our system to perform interactive information seeking. (A) *map view* colored for overview (B) *map view* colored for information seeking (C) *phrase view* (D) *document view*.

design tasks are formulated based on the following goals:

- **T1. Provide an informative overview of the PhraseMap.** Motivated by *R3.1*, the overview presentation is helpful only if a large amount of information is well-presented to users. Given the enormous amount of information in the PhraseMap, the system should avoid information clutter and simultaneously retain critical information.
- **T2. Offer goal-specific guidance to explore phrases of interest.** To enable efficient exploration over the PhraseMap, the visualization design is required to highlight phrases of interest. The guidance should also consider the specific goals of the users, e.g., a global exploration requires displaying a topic distribution of the whole corpus, whereas a local search based on users' input needs to highlight relevant phrases.
- **T3. Support intuitive interactions for visual analytics.** The system should provide intuitive interactions to enable efficient exploration of the PhraseMap, such as zooming and panning the PhraseMap and allowing users to filter and select phrases.
- **T4. Be responsive to users' feedback.** Knowledge exploration often contains multiple interactions between users and the system. The system must be intuitive for users to provide feedback and respond quickly.
- **T5. Support semantic zooming for knowledge exploration.** Text analysis involves multiple levels since users can first obtain a high-level idea and then study knowledge details. Besides, limited by the display area, visualizing all phrases without introducing visual clutter is almost impossible. The system should allow users to zoom in and out of the phrase map flexibly and semantically for multi-level analysis.
- **T6. Enable access to the raw information.** Original documents provide the most detailed information about concepts and the relations between concepts. The system should allow users to access raw documents after exploration.

5.4.2 User Interface

The system consists of three visual components and supports multiple operations for interaction. The main component is a *map view* to present the visualization of PhraseMap. It is a hexagonal grid layout in which each hexagon provides a space for displaying a phrase (**T5**). Since the view is uniformly partitioned, phrases displayed in respective hexagons would not occlude each other. Furthermore, phrases with similar semantics are located in similar positions to improve readability. Our system aggregates phrases into topics and assigns each topic a color to provide an overview of a corpus (**T1**). Besides, grids can also be colored by relevance to highlight relevant phrases (**T2**). The second component is a *phrase view*. It contains a list of phrases ranked by their relevance, and *accept/decline* buttons which allows users to provide feedback efficiently (**T3**). Finally, the third component is a *document view* for displaying the retrieved documents (**T6**). The three visual components are dynamically linked. Specifically, once users provide feedback through the *phrase view*, the two other views would update accordingly to the new relevance scores (**T4**). Also, users can click phrases on the *map view* to retrieve relevant documents in the *document view* (**T6**).

Here we illustrate how different views work together to support information seeking through a use case. Amy is a new researcher in the visualization field. She uses our system to retrieve useful knowledge from the VisPub dataset. Initially, key phrases extracted from the visualization literature are displayed on the *map view*. As shown in Figure 5A, Amy can efficiently discover the general topics of the literature. Amy then investigates the text visual analytic methods to locate the intriguing papers to read. She inputs a question "*what methods are proposed for text analysis?*" The *map view* updates immediately and highlights the phrases that could answer the question (Figure 5B). She can pan (), zoom (), and focus on () the *map view* to identify phrases of interest, and then click () a phrase to retrieve the documents and read them in the *document view* for studying detailed information (Figure 5D). Besides, in the *phrase view* (Figure 5C), the top-ranked phrases are displayed in descending order according to the relevance to Amy's question. She can select a set of phrases and click () the *accept* button () to inform the system of her interests. The relevance of the phrases and their neighbors to the question will increase. Similarly, if she clicks the decline button (), the relevance of related phrases will decrease. The recommended phrases will update immediately whenever phrases are accepted or declined.

5.4.3 Map View

The *map view* presents a grid-based layout for the PhraseMap, which organizes the extracted phrases in a semantic order. As introduced in the usage scenario, we design two color schemes to assist information seeking under two scenarios (1) displaying the global semantic structure when there is no query from users (Figure 6A). (2) coloring based on the relevance score when users specify the information of interest (Figure 6C).

Color for Overview. We use colors to help users differentiate semantic clusters. Specifically, we utilize a bottom-up clustering approach to group phrases iteratively – adjacent cells merge if their semantic distance is smaller than a user-defined threshold. Afterward, we assign each semantic cluster a random color in the palette. However, for the clusters that are too small, we assign them a background color () to prevent visual clutter. Besides, since cells are often more than phrases, a certain number of cells are empty. We colorize these cells in white. Finally, we draw a border between adjacent hexagons if they belong to distinct clusters to enhance readability.

In addition to colors, we display the *topic label* of each cluster at its centroid (Figure 6) to make the *map view* informative. To avoid using one single label to represent a large cluster, if the cluster size is larger than a threshold t (we set $t = 100$), we find all connected areas which share a common word and display the 2-gram or keyword sets as a topic label if the area size exceeds a threshold (we set t=60). For small clusters, we generate the topic label by selecting the 2-gram that appears the most in the cluster and use two top keywords instead if 2-gram is not available.

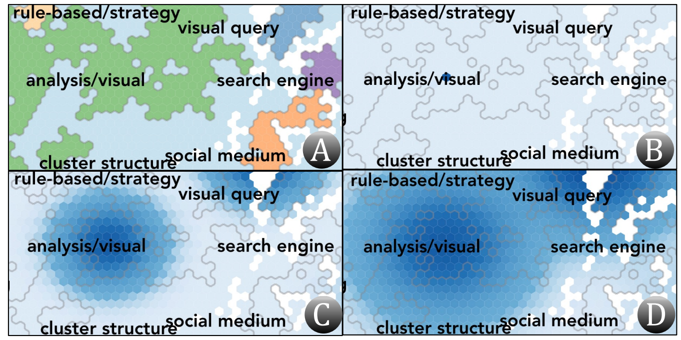


Fig. 6: Zoomed view of *map view*: (1) two color schema, colored for (A) overview and (B-D) information seeking (2) The influence of hyperparameters, i.e., bandwidth, (B) No KDE; (C) bandwidth=3; (D) bandwidth=5.

Color for Information Seeking. In response to users' feedback, the system also colors the phrases using the relevance scores to highlight recommended phrases. However, as illustrated in the Figure 6B, highlighting the top-ranked phrases alone cannot provide users with a satisfying visual experience because (1) phrases could be missed when discovering a large map, and (2) phrases that are semantically similar to the top-ranked ones should also be recommended. To address these issues, we highlight an area around recommended phrases by smoothing the relevance scores $\mathcal{S}$. Specifically, we perform kernel density estimation (KDE) to estimate the density of relevance score in each grid, with relevant phrases as samples and their relevance scores as weights. The density of each cell c is estimated by:

$$f(c) = \frac{1}{nh}\sum_{i=1}^{n} K(\frac{c - c_i}{h}), \tag{10}$$

where c_i is the center point of the sampled hexagon, n is the number of samples, K is the Gaussian kernel, and the bandwidth h is a smoothing parameter. The cells with high densities are in dark blue to indicate the relevant knowledge, as visualized in Figure 6C. We provide users with a slider bar to interactively change the bandwidth. Higher bandwidth results in highlighting a larger area of semantic similar phrases, as the comparison between $h = 3$ (Figure 6C) and $h = 5$ (Figure 6D).

5.4.4 Phrase View

Phrase view contains top-20 phrases ranked by the relevance score with two buttons (Figure 5C). It allows users to provide their feedback by clicking the phrases and the *accept* or *decline* button. It triggers the multi-modal navigation algorithm based on users' feedback. As a response to the user input, the other two views are also updated accordingly.

5.4.5 Document View

Document view presents a list of retrieved documents. Users can click each row to see content details. To support efficient information seeking, we also highlight the extracted phrases in each document, as shown in Figure 5D. The *document view* supports two levels of document retrieval from the other two views respectively: (1) **topic-level**: Once the user updates the relevance, the *document view* retrieves papers containing top-20 phrases shown in the *phrase* *view*; (2) **word-level**: users are allowed to click each phrase in the *map view* to retrieve specific papers in the *document view*.

5.4.6 Supported Follow-on Interactions

To support the efficient exploration of PhraseMap, our system also has a toolbox panel, which enables several follow-on interactions:
Semantic query. Users can type in a question in the search box as the input of the navigation algorithm. Then the color schema is automatically changed to color the relevance score.
Interactive & Semantic Zoom. To mimic the user experience of an actual map, we allow semantic zooming to show information at different granularity. Initially, we hide the details of each hexagon and only plot the semantic clusters with colors and boundaries, as visualized in Figure 7A. When users focus on (🔍) a specific region, we draw the hexagonal grids and display the text, which can be seen from Figure 7-b_2. Also, our system supports interactive zooming in/out (⤢) to demand detailed information.
Color Mode Switch. Users can freely switch between two color modes based on their needs. *Color for overview* is more helpful to provide a starting point for exploration, while *color for information seeking* provides more local details around users' interests.
Highlight and Tooltip. We highlight those whose relevance scores are high based on users' feedback to provide enough guidance over the graph. Also, when the user hovers (👆) the mouse over each phrase, we highlight the border of the corresponding hexagons and display the text information via a tooltip.

5.4.7 Implementation Detail

The interactive system is implemented with the Vue.js framework on the front end and a flask server at the back end. For the semantic query, it takes around three seconds to process 3300 abstracts in the VisPub. We provide details of the BERT model and RC-SOM training in the supplemental material.

6 Evaluation

To evaluate the performance of our visual analytic approach, we conducted three case studies using different academic datasets. Also, we conducted a comparison study to prove the usefulness of attention. Furthermore, we assess the RC-SOM's performance through quantitative and qualitative evaluations.

6.1 Datasets

We evaluated our PhraseMap on three datasets. (1) VisPub dataset: It contains 3504 IEEE Visualization publications from 1990-2021 [81] with three sub-areas: Scientific Visualization (SciVis), Information visualization (InfoVis), and Visual Analytics Science and Technology (VAST). (2) NeuraIPS dataset: 3116 documents were selected from the conference on Neural Information Processing Systems from 2017-2019 [1]. (3) COVID-19 dataset: It includes 2935 latest COVID papers. [2].

1. https://www.kaggle.com/datasets/rowhitswami/nips-papers-1987-2019-updated?select=papers.csv
2. https://www.kaggle.com/datasets/allen-institute-for-ai/CORD-19-research-challenge

6.2 Case Studies

Before case studies, we conducted a tutorial session with our domain experts, where we introduced the high-level concept and the details of the associated visual analytic system. It includes the purposes of each visualization, interaction component, and the supported functionalities. Based on domain experts' research interests and intentions, each expert chose a scientific paper dataset to explore with the system.

6.2.1 Information Seeking in the Visualization Field

Domain expert *E2* chose to explore general topics in the visualization literature and hence was shown the visualization in Figure 7. Because our layout makes semantically relevant clusters close to each other, three major groups can be seen, SciVis (●), InfoVis (●), and VAST (●), each of which was assigned a unique color. In each of the groups, representative keywords can be clearly identified. For example, "grid", "flow", "volume" are representative keywords of scientific visualization while "graph", "node", "layout" are relevant to the graph drawing problem, which is a subtopic of information visualization.

As described earlier, clusters are generated from a bottom-up clustering algorithm based on a pre-defined threshold of neighboring distance. By increasing the thresholds, adjacent cells with larger distances will be merged. It means more isolated phrases are connected, as can be seen from the comparison between Figure 7B, Figure 7C and Figure 7A. When manipulating the threshold, *E2* found three interesting evolution events: (1) *generation* of new groups, (2) *expansion* and (3) *merging* of existing groups. For the generation event, compared with Figure 7-b_1, two new topics pop up in Figure 7-c_1: "density"(●) and "uncertainty"(●). Take the "uncertainty" as an example, *E2* noticed some informative concepts related to the uncertainty when he zoomed in to the cell, including "uncertainty metric", "uncertainty flow", "uncertainty model" (Figure 7-c_4). When it comes to expansion, compared with the "3d" cluster (●) in Figure 7-b_1, it is clear to see that "2d" (●) in Figure 7-c_1 incorporates new knowledge, and is relevant to the concepts such as 2d map, as shown in Figure 7-c_3. As to merging, "sampling" (●) and "approximation" (●) in Figure 7-b_1 are merged into one group (●) with higher distance bound in Figure 7-c_1.

E2 performed information seeking with the question *what are the methods for particle data analysis*. From the resulting recommendations (Figure 9B_1), *E2* recognized these are all relevant methods to particle data, which can be grouped into two types. One is how to *render* the particle data, such as *volume rendering, context rendering technique, photon mapping*, and the other type is how to analyze particle tracing techniques, including *topological analysis, particle trace integration, etc.* By anchoring these phrases, *E2* confirmed that the retrieved papers are highly relevant to the analysis of particle data (Figure 9B_2). *E2* noticed that some distant phrases that belong to different clusters may retrieve the same paper. For example, in the paper: *case study: visualization of particle track data*, there is a sentence: " In this paper, we describe how techniques of *volume rendering* and *information visualization* are used to visualize the large particle trace data set generated from this high energy physics experiment". These two important phrases, volume rendering, and information visualization are accurately recommended from our PhraseMap, highlighted in the Figure 9B_2. *E2* commented that it would have taken a much longer time for users to identify these two semantically distant but relevant phrases without the recommendation from PhraseMap.

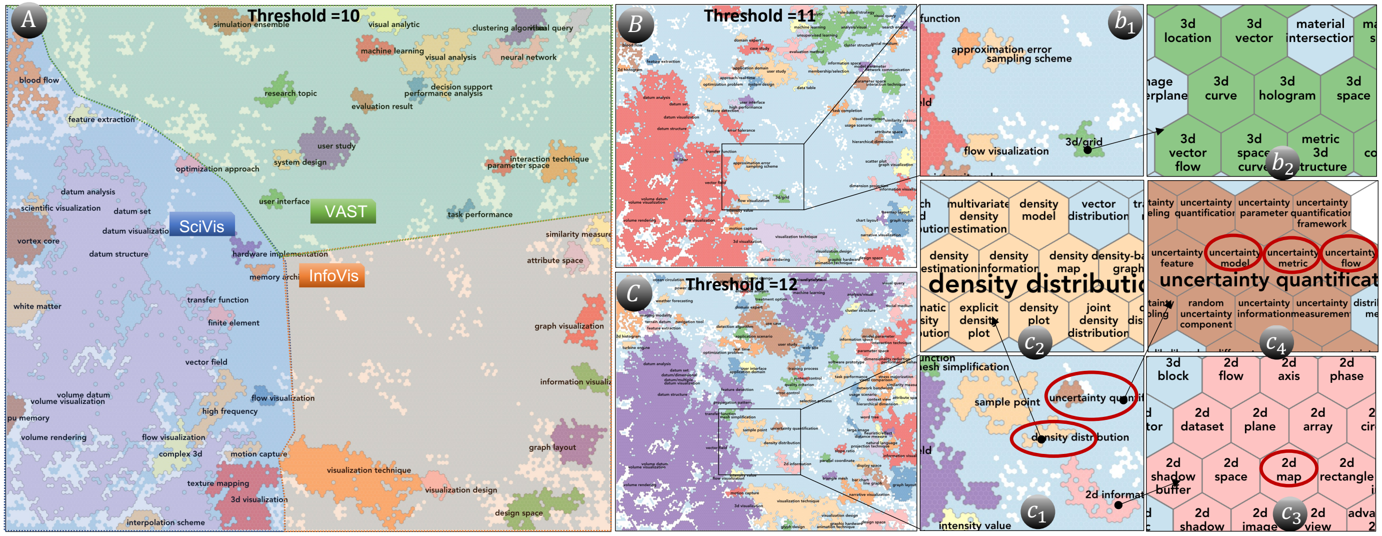


Fig. 7: (A-C) map view of the VisPub dataset with different thresholds. (b_1, c_1) zoomed-in views from the same position. (c_3) expanded cluster compared to b_2. (c_2) a newborn cluster of *density distribution*. (c_4) a newborn cluster of *uncertainty quantification*.

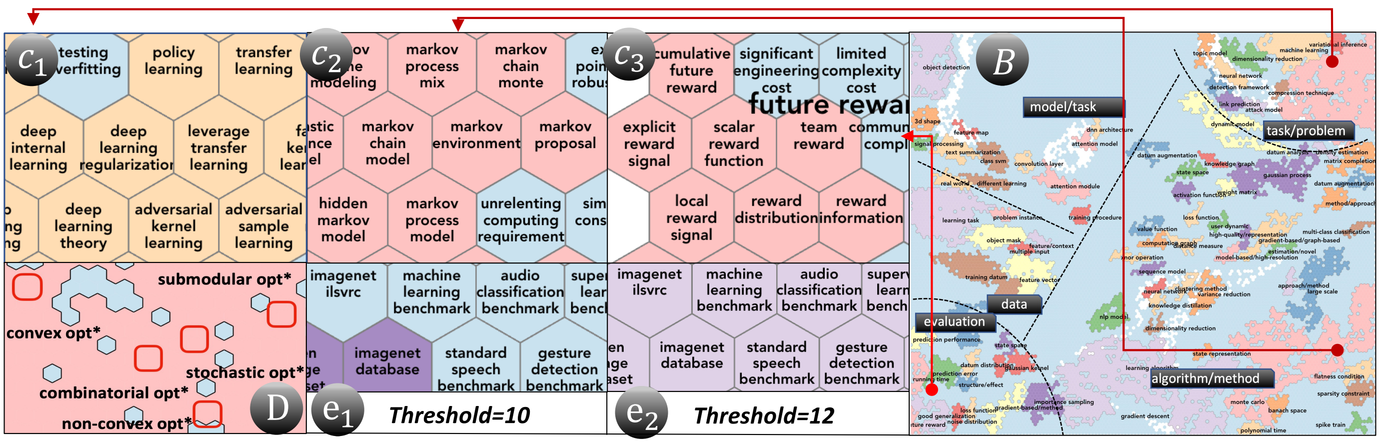


Fig. 8: (B) *map view* of the NeuraIPS dataset, zoomed view of clusters: (c_1) *transfer learning*, (c_2) *Markov chain*, (c_3) *reward function*. (D) cluster of *optimization*, ($e_1 - e_2$) The same zoomed view with different thresholds.

6.2.2 Cross-domain Analysis of ML Literature

We invited our domain expert *E3* to conduct a global exploration of this literature. As shown in the Figure 8B, it is clear to see some clusters with clear boundaries. *E3* checked each cluster and found they were corresponding to *data, algorithm/method, evaluation, task/problem* as labeled with tooltips in Figure 8B. When *E3* zoomed into the *task* cluster for details, she discovered that it can be further divided into finer-grained clusters, including general tasks such as classification, regression, optimization, and some application-specific tasks. Some tasks are relevant to image/video, such as recognition, segmentation, and tracking, while others are associated with natural language, e.g. machine translation, entity recognition, etc. After the exploration of different clusters, *E3* agreed that it is easy to identify similar phrases since they are close to each other in the map view. Meanwhile, *E3* was wondering whether the relationships among distant phrases are well captured in the grid layout. Specifically, *E3* pointed out that *reinforcement learning*(●), *Markov Chain*(●), *reward* (●) are related concepts in machine learning, and they are far away from each other in the grid layout due to the large distance in the semantic space. as shown in Figure 8$c_1 - c_3$. We later checked the edges in our attention-based graph and found there exist= multiple edges for each pairwise relation between them, take (reward, Markov Chain) as an example, one edge goes from *lipschitz reward function* to *observable Markov decision*, indicating the long-distance semantic relationships can be captured in our navigation algorithm based on the attention.

To get an overview of the literature, *E3* compared the map view with the 2019 NeuraIPS submission areas[3]. For many of the primary areas, *E3* can identify the corresponding clusters in Figure 8B, including *algorithms, optimization, probabilistic methods, reinforcement learning & planning, data & challenges*, seen from the cluster labels. Furthermore, *E3* found our system can support sub-area explorations in two ways: 1. Sub-areas are an important concept captured in one cluster. The NeuraIPS website shows the *optimization* area contains five sub-areas: convex, non-convex, submodular, combinatorial, and stochastic. *E3* found all matched phrases in the *optimization* cluster (Figure 8-*D*). It proves our PhraseMap can capture the important concepts from the corpus, as well as additional knowledge, e.g. *linear optimization* is not a domain sub-area but also contained in the cluster.

3. https://nips.cc/Conferences/2019/PaperInformation/SubjectAreas

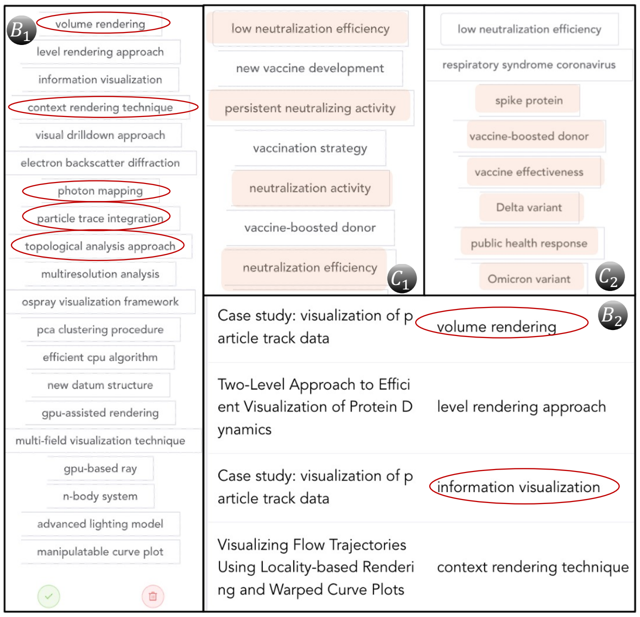


Fig. 9: (B_1) the initial recommendation of query: "what are the methods for particle data analysis", (B_2) relevant documents to *particle data*, (C_1-C_2) recommendation panels in the information-seeking process of the biomedical domain,

2. Sub-areas are adjacent small clusters, which can be grouped together if we increase the grouping threshold. According to the domain knowledge [3], *data & challenges* contains two sub-areas, *dataset* and *benchmarks*. We identify two corresponding clusters connected through the *ImageNet* (Figure 8e_1). They are merged into one cluster when we increase the threshold, shown in Figure 8e_2.

6.3 Interactive Retrieval of COVID-19 papers

To check whether the performance of our approach can support clinical practice, we generated the PhraseMap from 2935 latest COVID papers and invite *E1* to perform query-driven information seeking. *E1* is interested in "how efficient is the vaccination against the Delta and Omicron variants?" From the initial phrase recommendations, *E1* was surprised to discover a paper reporting the *low neutralization efficiency* of the third vaccination and expressed the necessity to read it [82]. Later, *E1* chose additional phrases that were relevant to *neutralization* from the initial recommendation. The updated recommendation is shown in Figure 9C_1, which ranked more *neutralization* related phrases higher, indicating the successful updating process steered by users' feedback. Following the search, *E1* has subsequent questions on whether the *neutralization* has different effects on different variants, and if there are any safety concerns related to vaccination. *E1* then included the phrase *vaccination breakthrough rate* and *vaccination rate*. As indicated in Figure 9C_2, the PhraseMap successfully returned two important variants, the *Delta variant*, and *Omicron variant* as specified in the original query. This also indicates that our QA model can accurately identify semantically relevant phrases as relevant knowledge. Through the system, *E1* was able to locate several highly-relevant documents answering the effects of vaccination on different variants [83], [84]. In addition to vaccination, *E2* also finds other documents retrieved by *public health response* propose different methods to identify VoC (variants of concern), which provides early warning to communities. For example, Yang et al. incorporate viral replication dynamics into the evaluation of the SARS-CoV-2 variant impact on immunity and VE [85]. In the end, *E1* commented on the high efficiency of locating relevant documents in a few steps without a lengthy navigating and seeking process. Also, it can not only answer the user's query but also recommends more relevant information for users to explore.

6.4 Contextual Attention vs. Semantic Similarity

We construct a PhraseMap by modeling the relationships of contextual phrases. Each edge on the PhraseMap indicates the attention of a phrase to/from its neighboring phrase in documents. Besides the attention to contextual phrases, another approach to constructing the PhraseMap is based on the semantic similarity of phrases. Specifically, we apply the pre-trained BERT model to project each phrase into a latent space, and then compute the cosine similarity of latent vectors to determine the phrase similarity.

We evaluated the contextual attention and semantic similarity by comparing the recommended phrases during information seeking. In the comparison, the VisPub dataset was used to construct the PhraseMaps, and several representative phrases in the visualization field were selected. The phrases cover three aspects: data type, application area, and algorithm. We fed the phrases into the navigation algorithm and compared the top-5 recommended phrases.

Results and Discussions. Table 2 shows the top-5 recommended phrases that were related to the given phrases. As can be seen, most phrases recommended using semantic similarity were redundant. For example, when users clicked the phrase "topic modeling", the PhraseMap constructed using semantic similarity would refer users to "topic modeling method", "topic modeling technique", and "topic model refinement", which do not provide more information than "topic modeling". Similar phenomenons appeared when other phrases were clicked, e.g., "domain expert". In addition, remind that our method displays semantically similar phrases at nearby positions. Visual exploration can easily identify such phrases, and the recommendation is unnecessary. By contrast, the PhraseMap constructed based on contextual attention of phrases does not suffer from the problems mentioned above. When users clicked the phrase "topic modeling", the PhraseMap referred them to "text document collection", "vector space model", and "entity recognition technique". The three terms could help users understand how data for topic modeling were collected, what models were used, and how entities in the text were recognized.

We show two additional examples to demonstrate the effectiveness of contextual attention over the semantic similarity in phrase recommendation. First, if users were interested in a specific type of data, the PhraseMap built upon contextual attention would recommend methods that can process such data type. For example, given the phrase "vector field", it refers readers to the "flow topology method" and "fluid flow visualization". Second, if users know only a high-level concept, the attention would recommend phrases related to concrete ideas. An example of this is "domain expert". The contextual attention recommends "event sequence exploration", "business competition", and "car engine design", which were applications requiring domain expertise for evaluation. Note that the PhraseMap constructed based on the semantic similarity of phrases could not achieve the advantages. Please refer to Table 2 for the comparison results and more examples.

Efficient Document Retrieval. Although our PhraseMap can recommend diverse and highly relevant phrases to users' interests,

Topic	Accepted	Met.	Rank1	Rank2	Rank3	Rank4	Rank5
Data Type	*point data*	*sim*	data set	data point	data display	data analysis	volume rendering
		att	intercluster galaxy	horologium-reticulum supercluster	voxel grid	3d galaxy position	point particle
	vector field	*sim*	tensor field	input vector field	3d vector field	vector field sample	vector field
		att	3d data set	flow topology method	fluid flow visualization	tensor field visualization	data access prediction
Domain Area	*domain expert*	*sim*	domain expert group	domain specialist	domain expertise	domain user	domain scientist
		att	real-world example	evenet sequence exploration	business competition environment	car engine design	data modality
	graph analysis application	*sim*	analytic tool	analysis task	analysis process	analysis framework	analysis shift
		att	graph visualization application	graph visualization technique	analysis task	node-link visualization	network embedding
Method	*topic modeling*	*sim*	topic modeling method	topic modeling technique	topic model refinement	topic model quality	text analysis
		att	text document collection	vector space model	entity recognition technique	loop process	text analytic
	machine learning algorithm	*sim*	machine learning technique	machine learning	machine learning system	machine learning method	machine learning process
		att	machine learning	hypothesis-based evaluation	error experiment	ml fairness metric	analytic tool

TABLE 2: A comparison study of navigation algorithm on the PhraseMap with two different measurements (i.e., *metric*): semantic similarity of phrase embeddings (i.e., *sim*) and attention scores between two phrases (i.e., *att*). Column of *topic* indicates the category of *phrases* we select as positive feedback in the navigation algorithm.

the phrases still lack sufficient details for studying knowledge. We locate the documents where the given phrase and the recommended phrases co-exist for users to read. For example, documents containing "entity recognition technique" and "topic modeling" may describe the way to identify events in the news and social media [86]; and the document containing "vector space model" and "topic modeling" explains how users adjust the topics semantically.

6.5 Evaluation of Resource Controlled-SOM

We apply the RC-SOM to display phrases on a grid layout. We evaluate it both qualitatively and quantitatively.

t-SNE vs. RC-SOM We qualitatively compared the RC-SOM and scatterplot results to evaluate this grid-based visualization. In the comparison, we applied the t-SNE to project BERT embeddings into 2D space and show the projected phrases using the scatterplot. As indicated in Figures 10-a_1 and a_2, the global structures of the phrases revealed by the RC-SOM and the scatterplot are similar. The results imply that the RC-SOM can well present the semantic information hidden in BERT embeddings, although RC-SOM also expects to utilize the displaying area well. When users zoom in to the scatterplot and attempt to read the phrases, letters overlap with each other, and phrases are difficult to read. Only the phrases *"overdraw problem"*, *"fluid viscosity"*, and *"low compression"* can be recognized in Figure 10-b_1. By contrast, all phrases are readable in the layout generated by our RC-SOM (Figure 10-b_2).

RC-SOM vs. SOM We compare the space utilization of our RC-SOM with the original SOM. Two well-known datasets with ground truth labels are chosen for the quantitative comparison: MNIST [87] and IRIS [88]. We project high-dimensional data to 2D space using two methods and colorize the results according to their labels to achieve a visual comparison. Due to the limited space, we sampled $1,000$ images from the MNIST dataset with 32×32 grids. For the IRIS dataset, we apply 13×13 to display all 150 items. The strategy ensures that the hexagon grid has enough capacity to show all data items.

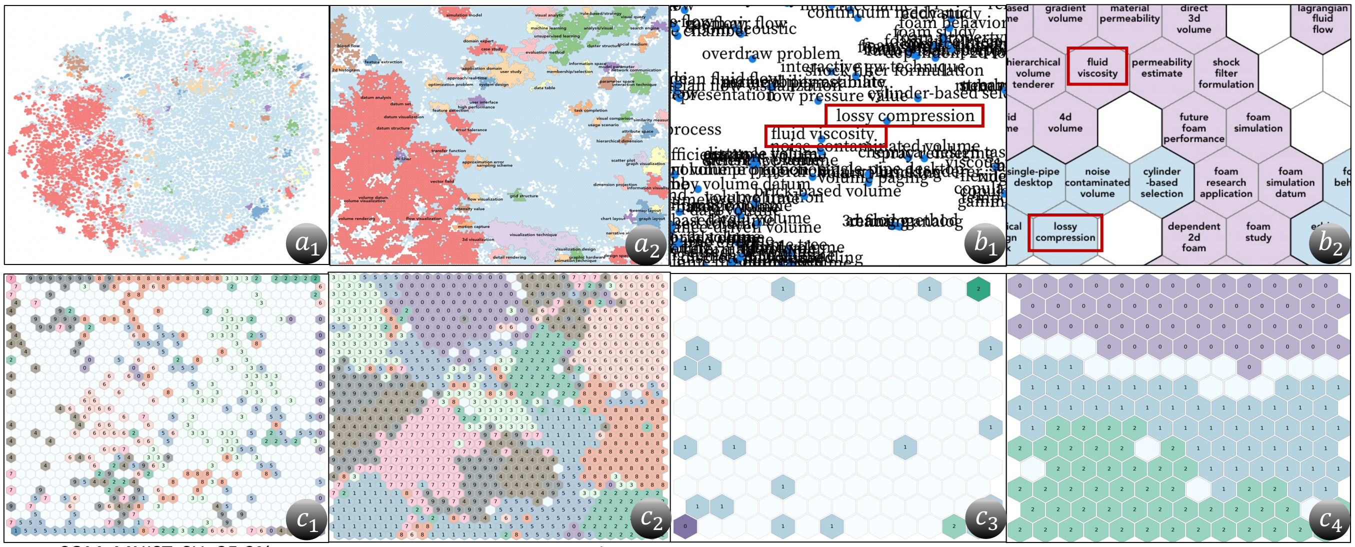
As shown in the Figure 10, our RC-SOM displays only one item, and neighboring cells display similar items. Therefore, it exhibits a clear cluster pattern, well utilizes the space, and arranges items neatly in the visualization space. It achieves 97.7% (c_2) and 88.9% (c_4) of space utilization for two datasets, which has been increased a lot from the original SOM ($c_{1\&3}$).

7 Expert Feedback and Discussion

Expert Feedback We interviewed our domain experts after the case studies. Overall, they were satisfied with the system and the navigation functionality during the information-seeking process. They were particularly pleased with the display of PhraseMap and the ability to retrieve information through query and sampling-based recommendations. *E1* commented that the recommended phrases after the *accepting* operation mimics the human's mental attention mechanism in associating relevant concepts. They also enjoyed the experience of exploring the PhraseMap, similar to exploring a topographical map.

The experts also provided constructive feedback for improvements. E3 commented on the semantic query function and said, "*it would be interesting if the user feedback could be collected as training data and utilized to fine-tune the model.*" It is feasible if we collect the questions and users' final phrases set as the ground truth to fine-tune the QA model. Nevertheless, the model's performance is hard to predict given that this is an open-ended task. E2 added "*the user experience will be improved if the response time is reduced.*" We agree that our current relevance score computation, as reported in Section 5.4.7, will be faster if we first locate a subgraph of relevant phrases without computing the relevance score of all phrases to the query. Moreover, E1 expressed that it is helpful that PhraseMap provides topics for areas with similar phrases.

Limitations There are several limitations exposed during the case studies. First, to fully support interactive exploration with PhraseMap, additional functionalities in the supporting views can be further incorporated. For example, adding mapping between the phrases in the recommendation panel and the highlighted areas for quick localization without zooming and dragging could make the interaction more intuitive and efficient. As for the document view, we can present more document details to users if users are interested.



SOM, *MNIST*, SU: 35.2% *RC-SOM*, *MNIST*, **SU: 97.7%** *SOM*, *IRIS*, SU: 12.4% *RC-SOM*, *IRIS*, **SU: 88.9%**

Fig. 10: The comparison of t-SNE and RC-SOM in ($a_{1\&2}$) global views and ($b_{1\&2}$) local views. ($c_1 - c_4$) the space utilization (SU) between SOM and RC-SOM using the MNIST and IRIS datasets.

Second, the lack of training datasets limits the generalizability of our approach in the domain that lacks QA fine-tuning datasets. However, it is still possible and promising to perform transfer learning by fine-tuning multiple sequential tasks, similar to what we achieved for the QA model in the visualization literature. It can be easily addressed since the pre-training language models only need a small set of labeled data to fine-tune.

Future Work Given the extensive information in the PhraseMap, it is interesting to apply graph analysis for more tasks, such as identifying important phrases and automatically generating hierarchical semantic clusters. Also, since the relationships are built from the papers in different years, the PhraseMap can explicitly add temporal information and enable trend evolution analysis by dynamic graph layouts. Moreover, we are also interested in extending PhraseMap into a knowledge graph by enriching the relationships. It can support more intelligent exploration with the help of NLP techniques.

8 Conclusion

In this work, we propose to utilize the attention mechanism in the language model to construct a phrase-based graph, named PhraseMap. Due to the extensive volume of information in the PhraseMap, we introduce a navigation algorithm to facilitate efficient information seeking of the PhraseMap. It includes a question-answering (QA) model to compute the relevance score of phrases and an information propagation algorithm to update the scores based on the user's feedback. To better present the PhraseMap, we develop a resource-controlled self-organizing map (RC-SOM) algorithm and implement a visual analytic system. Through evaluation and case studies, we demonstrate the usefulness and effectiveness of the approach.

9 Acknowledgements

This work is financially supported by NSF under grant number OAC-2112606.

References

[1] A. Barratt, "Evidence based medicine and shared decision making: The challenge of getting both evidence and preferences into health care," *Patient education and counseling*, vol. 73, no. 3, pp. 407–412, 2008.

[2] T. Hofmann, "Probabilistic latent semantic indexing," in *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '99, Berkeley, California, USA: Association for Computing Machinery, 1999, pp. 50–57, ISBN: 1581130961. DOI: 10.1145/312624.312649.

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, no. null, pp. 993–1022, Mar. 2003, ISSN: 1532-4435.

[4] D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in Neural Information Processing Systems*, T. Leen, T. Dietterich, and V. Tresp, Eds., vol. 13, MIT Press, 2000.

[5] S. Robertson and H. Zaragoza, "The probabilistic relevance framework: Bm25 and beyond," *Found. Trends Inf. Retr.*, vol. 3, no. 4, pp. 333–389, Apr. 2009, ISSN: 1554-0669. DOI: 10.1561/1500000019.

[6] X. Wang, A. McCallum, and X. Wei, "Topical n-grams: Phrase and topic discovery, with an application to information retrieval," in *Seventh IEEE international conference on data mining (ICDM 2007)*, IEEE, 2007, pp. 697–702.

[7] D. M. Blei and J. D. Lafferty, "Visualizing topics with multi-word expressions," *arXiv preprint arXiv:0907.1013*, 2009.

[8] Y. Kim, M. Kim, A. Cattle, J. Otmakhova, S. Park, and H. Shin, "Applying graph-based keyword extraction to document retrieval," in *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, 2013, pp. 864–868.

[9] J. Lee, M. Sung, J. Kang, and D. Chen, "Learning dense representations of phrases at scale," in *Association for Computational Linguistics (ACL)*, 2021.

[10] S. Wang, L. Thompson, and M. Iyyer, "Phrase-bert: Improved phrase embeddings from bert with an application to corpus exploration," *arXiv preprint arXiv:2109.06304*, 2021.

[11] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, *Albert: A lite bert for self-supervised learning of language representations*, 2019. DOI: 10.48550/ARXIV.1909.11942.

[12] Y. Liu, M. Ott, N. Goyal, *et al.*, *Roberta: A robustly optimized bert pretraining approach*, 2019. DOI: 10.48550/ARXIV.1907.11692.

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

[14] D. Fried and S. G. Kobourov, "Maps of computer science," in *2014 IEEE Pacific Visualization Symposium*, IEEE, 2014, pp. 113–120.

[15] R. Burd, K. A. Espy, M. I. Hossain, S. Kobourov, N. Merchant, and H. Purchase, "Gram: Global research activity map," in *Proceedings of the 2018 International Conference on Advanced Visual Interfaces*, 2018, pp. 1–9.

[16] K. Borner, *Atlas of science: Visualizing what we know*. Mit Press, 2010.

[17] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, “Transformers in vision: A survey,” *ACM Computing Surveys (CSUR)*, 2021.

[18] S. Zhang, L. Yao, A. Sun, and Y. Tay, “Deep learning based recommender system: A survey and new perspectives,” *ACM Computing Surveys (CSUR)*, vol. 52, no. 1, pp. 1–38, 2019.

[19] A. Galassi, M. Lippi, and P. Torroni, “Attention in natural language processing,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 10, pp. 4291–4308, 2020.

[20] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.

[21] T. Mikolov, M. Karafiát, L. Burget, J. Cernock, and S. Khudanpur, “Recurrent neural network based language model.,” in *Interspeech*, Makuhari, vol. 2, 2010, pp. 1045–1048.

[22] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, *et al.*, “Improving language understanding by generative pre-training,” 2018.

[23] D. Mareček and R. Rosa, “From balustrades to pierre vinken: Looking for syntax in transformer self-attentions,” in *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 263–275. DOI: 10.18653/v1/W19-4827.

[24] O. Kovaleva, A. Romanov, A. Rogers, and A. Rumshisky, “Revealing the dark secrets of BERT,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 4365–4374. DOI: 10.18653/v1/D19-1445.

[25] E. Reif, A. Yuan, M. Wattenberg, *et al.*, “Visualizing and measuring the geometry of bert,” in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32, Curran Associates, Inc., 2019.

[26] Q. Xu, L. Zhu, T. Dai, and C. Yan, “Aspect-based sentiment classification with multi-attention network,” *Neurocomputing*, vol. 388, pp. 135–143, 2020.

[27] D. Tang, B. Qin, and T. Liu, “Aspect level sentiment classification with deep memory network,” *arXiv preprint arXiv:1605.08900*, 2016.

[28] Y. Cui, Z. Chen, S. Wei, S. Wang, T. Liu, and G. Hu, “Attention-over-attention neural networks for reading comprehension,” *arXiv preprint arXiv:1607.04423*, 2016.

[29] D. Ma, S. Li, X. Zhang, and H. Wang, “Interactive attention networks for aspect-level sentiment classification,” *arXiv preprint arXiv:1709.00893*, 2017.

[30] H. Zuo, Y. Yin, and P. Childs, “Patent-kg: Patent knowledge graph extraction for engineering design,” *Proceedings of the Design Society*, vol. 2, pp. 821–830, 2022.

[31] T. Kim, Y. Yun, and N. Kim, “Deep learning-based knowledge graph generation for covid-19,” *Sustainability*, vol. 13, no. 4, p. 2276, 2021.

[32] Y. Tu, J. Xu, and H.-W. Shen, “Keywordmap: Attention-based visual exploration for keyword analysis,” in *2021 IEEE 14th Pacific Visualization Symposium (PacificVis)*, IEEE, 2021, pp. 206–215.

[33] G. L. D. l. P. Sarracén and P. Rosso, “Offensive keyword extraction based on the attention mechanism of bert and the eigenvector centrality using a graph representation,” *Personal and Ubiquitous Computing*, pp. 1–13, 2021.

[34] H. Ding and X. Luo, “Attention-based unsupervised keyphrase extraction and phrase graph for covid-19 medical literature retrieval,” *ACM Transactions on Computing for Healthcare (HEALTH)*, vol. 3, no. 1, pp. 1–16, 2021.

[35] N. Cao and W. Cui, “Introduction to text visualization. berlin, germany,” Springer.

[36] M. Berger, K. McDonough, and L. M. Seversky, “Cite2vec: Citation-driven document exploration via word embeddings,” *IEEE transactions on visualization and computer graphics*, vol. 23, no. 1, pp. 691–700, 2016.

[37] M. Brehmer, S. Ingram, J. Stray, and T. Munzner, “Overview: The design, adoption, and analysis of a visual document mining tool for investigative journalists,” *IEEE transactions on visualization and computer graphics*, vol. 20, no. 12, pp. 2271–2280, 2014.

[38] J. Choo, C. Lee, C. K. Reddy, and H. Park, “Utopian: User-driven topic modeling based on interactive nonnegative matrix factorization,” *IEEE transactions on visualization and computer graphics*, vol. 19, no. 12, pp. 1992–2001, 2013.

[39] F. Heimerl, M. John, Q. Han, S. Koch, and T. Ertl, “Docucompass: Effective exploration of document landscapes,” in *2016 IEEE conference on visual analytics science and technology (VAST)*, IEEE, 2016, pp. 11–20.

[40] K. Pearson, “Liii. on lines and planes of closest fit to systems of points in space,” *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, vol. 2, no. 11, pp. 559–572, 1901.

[41] T. Kohonen and T. Honkela, “Kohonen network,” *Scholarpedia*, vol. 2, no. 1, p. 1568, 2007.

[42] T. Kohonen, “Self-organized formation of topologically correct feature maps,” *Biological cybernetics*, vol. 43, no. 1, pp. 59–69, 1982.

[43] L. Van der Maaten and G. Hinton, “Visualizing data using t-sne.,” *Journal of machine learning research*, vol. 9, no. 11, 2008.

[44] L. McInnes, J. Healy, and J. Melville, “Umap: Uniform manifold approximation and projection for dimension reduction,” *arXiv preprint arXiv:1802.03426*, 2018.

[45] E. A. Bier, S. K. Card, and J. W. Bodnar, “Principles and tools for collaborative entity-based intelligence analysis,” *IEEE transactions on visualization and computer graphics*, vol. 16, no. 2, pp. 178–191, 2009.

[46] M. S. Hossain, P. Butler, A. P. Boedihardjo, and N. Ramakrishnan, “Storytelling in entity networks to support intelligence analysts,” in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2012, pp. 1375–1383.

[47] S. Liu, X. Wang, C. Collins, *et al.*, “Bridging text visualization and mining: A task-driven survey,” *IEEE transactions on visualization and computer graphics*, vol. 25, no. 7, pp. 2482–2504, 2018.

[48] A. Skupin and S. I. Fabrikant, “Spatialization methods: A cartographic research agenda for non-geographic information visualization,” *Cartography and Geographic Information Science*, vol. 30, no. 2, pp. 99–119, 2003.

[49] M. Liu, S. Liu, X. Zhu, Q. Liao, F. Wei, and S. Pan, “An uncertainty-aware approach for exploratory microblog retrieval,” *IEEE transactions on visualization and computer graphics*, vol. 22, no. 1, pp. 250–259, 2015.

[50] X. Wang, S. Liu, Y. Chen, *et al.*, “How ideas flow across multiple social groups,” in *2016 IEEE Conference on Visual Analytics Science and Technology (VAST)*, IEEE, 2016, pp. 51–60.

[51] K. Andrews, W. Kienreich, V. Sabol, *et al.*, “The infosky visual explorer: Exploiting hierarchical structure and document similarities,” *Information Visualization*, vol. 1, no. 3-4, pp. 166–181, 2002.

[52] A. Nocaj and U. Brandes, “Organizing search results with a reference map,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 12, pp. 2546–2555, 2012.

[53] A. Debiasi, “Study of visual clutter in geographic node-link diagrams,” Ph.D. dissertation, University of Trento, 2016.

[54] E. R. Gansner, Y. Hu, and S. G. Kobourov, “Gmap: Drawing graphs as maps,” in *International Symposium on Graph Drawing*, Springer, 2009, pp. 405–407.

[55] S. Chen, S. Chen, Z. Wang, *et al.*, “D-map: Visual analysis of ego-centric information diffusion patterns in social media,” in *2016 IEEE Conference on Visual Analytics Science and Technology (VAST)*, IEEE, 2016, pp. 41–50.

[56] H. Couclelis, “Worlds of information: The geographic metaphor in the visualization of complex information,” *Cartography and geographic information systems*, vol. 25, no. 4, pp. 209–220, 1998.

[57] S. Chen, S. Chen, Z. Wang, J. Liang, Y. Wu, and X. Yuan, “D-map+ interactive visual analysis and exploration of ego-centric and event-centric information diffusion patterns in social media,” *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 1, pp. 1–26, 2018.

[58] S. Chen, S. Li, S. Chen, and X. Yuan, “R-map: A map metaphor for visualizing information reposting process in social media,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 1, pp. 1204–1214, 2019.

[59] D. Lewis and K. Jones, “Natural language processing for information retrieval,” *Communications of the ACM*, vol. 39, Dec. 1996. DOI: 10.1145/234173.234210.

[60] K. Nishida, I. Saito, A. Otsuka, H. Asano, and J. Tomita, “Retrieve-and-read: Multi-task learning of information retrieval and reading comprehension,” in *Proceedings of the 27th ACM international conference on information and knowledge management*, 2018, pp. 647–656.

[61] C. Qu, L. Yang, M. Qiu, W. B. Croft, Y. Zhang, and M. Iyyer, “Bert with history answer embedding for conversational question answering,” in *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*, 2019, pp. 1133–1136.

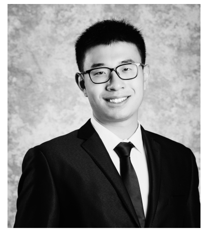
[62] Z. Li, C. Zhang, S. Jia, and J. Zhang, “Galex: Exploring the evolution and intersection of disciplines,” *IEEE transactions on visualization and computer graphics*, vol. 26, no. 1, pp. 1182–1192, 2019.

[63] S. Liu, X. Wang, J. Chen, J. Zhu, and B. Guo, “Topicpanorama: A full picture of relevant topics,” in *2014 IEEE Conference on Visual Analytics Science and Technology (VAST)*, IEEE, 2014, pp. 183–192.

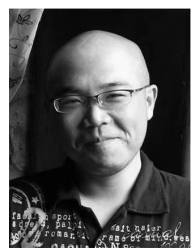
[64] W. Yang, X. Wang, J. Lu, W. Dou, and S. Liu, “Interactive steering of hierarchical clustering,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 10, pp. 3953–3967, 2021.

[65] D. Park, S. Kim, J. Lee, J. Choo, N. Diakopoulos, and N. Elmqvist, "Conceptvector: Text visual analytics via interactive lexicon building using word embedding," *IEEE transactions on visualization and computer graphics*, vol. 24, no. 1, pp. 361–370, 2017.
[66] I. Beltagy, K. Lo, and A. Cohan, "Scibert: A pretrained language model for scientific text," *arXiv preprint arXiv:1903.10676*, 2019.
[67] J. Lee, W. Yoon, S. Kim, *et al.*, "Biobert: A pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, 2020.
[68] A. Rogers, O. Kovaleva, and A. Rumshisky, "A primer in bertology: What we know about how bert works," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 842–866, 2020.
[69] I. Tenney, D. Das, and E. Pavlick, "Bert rediscovers the classical nlp pipeline," *arXiv preprint arXiv:1905.05950*, 2019.
[70] L. Cui, S. Cheng, Y. Wu, and Y. Zhang, "Does bert solve commonsense task via commonsense knowledge," *arXiv preprint arXiv:2008.03945*, vol. 4, 2020.
[71] G. Jawahar, B. Sagot, and D. Seddah, "What does bert learn about the structure of language?" In *ACL 2019-57th Annual Meeting of the Association for Computational Linguistics*, 2019.
[72] P. M. Vu, H. V. Pham, T. T. Nguyen, and T. T. Nguyen, "Phrase-based extraction of user opinions in mobile app reviews," in *Proceedings of the 31st IEEE/ACM international conference on automated software engineering*, 2016, pp. 726–731.
[73] B. Santorini, "Part-of-Speech Tagging Guidelines for the Penn Treebank Project 3rd Revision, 2nd printing," en, p. 34,
[74] T. Eftimov, B. Koroušić Seljak, and P. Korošec, "A rule-based named-entity recognition method for knowledge extraction of evidence-based dietary recommendations," *PloS one*, vol. 12, no. 6, e0179488, 2017.
[75] N. Perera, M. Dehmer, and F. Emmert-Streib, "Named entity recognition and relation detection for biomedical information extraction," *Frontiers in cell and developmental biology*, p. 673, 2020.
[76] B. Waltl, G. Bonczek, and F. Matthes, "Rule-based information extraction: Advantages, limitations, and perspectives," *Jusletter IT (02 2018)*, 2018.
[77] Y. Y. Leow, T. Laurent, and X. Bresson, "Graphtsne: A visualization technique for graph-structured data," *arXiv preprint arXiv:1904.06915*, 2019.
[78] D. C. Zhang and H. W. Lauw, "Semi-supervised semantic visualization for networked documents," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, 2021, pp. 762–778.
[79] E. Berglund and J. Sitte, "The parameterless self-organizing map algorithm," *IEEE Transactions on neural networks*, vol. 17, no. 2, pp. 305–316, 2006.
[80] T. Kohonen, *Self-organizing maps*. Springer Science & Business Media, 2012, vol. 30.
[81] P. Isenberg, F. Heimerl, S. Koch, *et al.*, "vispubdata.org: A Metadata Collection about IEEE Visualization (VIS) Publications," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, 2017, To appear. DOI: https://doi.org/10.1109/TVCG.2016.2615308.
[82] I. Nemet, L. Kliker, Y. Lustig, *et al.*, "Third bnt162b2 vaccination neutralization of sars-cov-2 omicron infection," *New England Journal of Medicine*, vol. 386, no. 5, pp. 492–494, 2022.
[83] M. Hoffmann, N. Krüger, S. Schulz, *et al.*, "The omicron variant is highly resistant against antibody-mediated neutralization: Implications for control of the covid-19 pandemic," *Cell*, vol. 185, no. 3, pp. 447–456, 2022.
[84] Y. Liu, N. Zhang, B. Wang, and Y. Zhang, "Broad and long-lasting immune response against sars-cov-2 omicron and other variants by pika-adjuvanted recombinant sars-cov-2 spike (s) protein subunit vaccine (ys-sc2-010)," *bioRxiv*, 2021.
[85] W. Yang and J. Shaman, "Viral replication dynamics could critically modulate vaccine effectiveness and should be accounted for when assessing new sars-cov-2 variants," *Influenza and Other Respiratory Viruses*, vol. 16, no. 2, p. 366, 2022.
[86] W. Dou, X. Wang, D. Skau, W. Ribarsky, and M. X. Zhou, "Leadline: Interactive visual analysis of text data through event identification and exploration," in *2012 IEEE Conference on Visual Analytics Science and Technology (VAST)*, IEEE, 2012, pp. 93–102.
[87] L. Deng, "The mnist database of handwritten digit images for machine learning research," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.
[88] R. A. FISHER, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, no. 2, pp. 179–188, 1936. DOI: https://doi.org/10.1111/j.1469-1809.1936.tb02137.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1469-1809.1936.tb02137.x.

Yamei Tu received a BS degree in Software Engineering from East China Normal University. She is a Ph.D. student in the GRAphics & VIsualization sTudY (GRAVITY) Research Group under Computer Science and Engineering at the Ohio State University. Her research interests are Visualization, Text Analysis, and Machine Learning.



Rui Qiu received the BS degree in Mathematics from Tianjin University of Finance and Economics, and the Master (2019) degree in Financial Math from Washington University in St. Louis. He is currently a PhD student in Computer Science and Engineering with The Ohio State University. His research interests include visual text analysis, information retrieval, NLP and clinical informatics.



Yu-Shuen Wang is an associate professor of the Department of Computer Science at National Chiao-Tung University. He received his PhD degree from Visual System Laboratory, National Cheng Kung University, Tainan, Taiwan, ROC, in 2010. Currently, he leads the Computer Graphics and Visualization Lab at the Institute of Multimedia Engineering. Prof. Wang's research interests include Computer Graphics, Image Manipulation, Map Design, Data Visualization, Human Computer Interface and Virtual Reality.

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 assistant 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 humancomputer 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, the MS degree in computer science from the State University of New York at Stony Brook, and the PhD degree in computer science from the University of Utah. He is a full professor with the Ohio State University. His primary research interests include scientific visualization and computer graphics.